Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2012

Samuel de Sousa Silva

Análise Funcional do Ventrículo Esquerdo em Angio–TC Coronária

Left Ventricle Functional Analysis from Coronary CT Angiography

Universidade de Aveiro
2012

Departamento de Electrónica, Telecomunicações e Informática

Samuel de Sousa Silva

Análise Funcional do Ventrículo Esquerdo em Angio–TC Coronária

Left Ventricle Functional Analysis from Coronary CT Angiography

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Informática, realizada sob a orientação científica do Dr.-Ing. Joaquim João Estrela Ribeiro Silvestre Madeira, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro, e da Doutora Maria Beatriz Alves de Sousa Santos, Professora Associada com Agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Apoio financeiro da FCT e do FSE no âmbito do III Quadro Comunitário de Apoio.

À deusa dos olhos garços, por me guiar até casa.

o júri

presidente

Doutora Maria Ana Dias Monteiro Santos
Professora Catedrática do Departamento de Biologia da Universidade de Aveiro

Doutor Aurélio Joaquim de Castro Campilho
Professor Catedrático da Faculdade de Engenharia da Universidade do Porto

Doutora Maria Beatriz Alves de Sousa Santos
Professora Associada com Agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro (Coorientadora)

Doutor Manuel João Toscano Próspero dos Santos
Professor Associado da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Dr.-Ing. Joaquim João Estrela Ribeiro Silvestre Madeira
Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro (Orientador)

Doutor João Manuel Duarte Cunha
Investigador Coordenador do LNEC (aposentado) – Laboratório Nacional de Engenharia Civil

agradecimentos

O presente trabalho é fruto da colaboração de inúmeras pessoas e entidades a quem devo um agradecimento.

Antes de mais, um profundo agradecimento ao Serviço de Cardiologia do Centro Hospitalar de Vila Nova de Gaia/Espinho (CHVNG/E), e em particular ao Doutor Vasco Gama, por ter proporcionado o cenário sobre o qual o trabalho pôde ser desenvolvido. Um agradecimento especial ao Doutor Nuno Bettencourt e aos técnicos de radiologia Mónica Carvalho, Daniel Leite e João Rocha pela disponibilidade constante, pela amabilidade e paciência e pela interessante troca de ideias. Sem a contínua colaboração deles, nada do que aqui é apresentado teria sido possível.

Um agradecimento é também devido aos meus orientadores, o Professor Joaquim Madeira e a Professora Beatriz de Sousa Santos, pelo acompanhamento que fizeram deste trabalho e por me terem proporcionado completa liberdade na prossecução das diferentes ideias que foram surgindo. Agradeço também ao Professor Carlos Ferreira toda a importante ajuda que prestou ao longo destes anos.

Uma palavra de agradecimento ao Professor Augusto Silva por ter propiciado o primeiro contacto com o Serviço de Cardiologia do CHVNG/E.

O Instituto de Engenharia Electrónica e Telemática de Aveiro (IEETA) teve também um papel importante no trabalho desenvolvido, proporcionando um ambiente de trabalho agradável e dinâmico.

A Fundação para a Ciência e Tecnologia (FCT) foi também crucial em todo o processo através da atribuição de uma bolsa de doutoramento que suportou o meu trabalho ao longo dos quatro anos que ele durou.

Não menos importantes agradecimentos são devidos a um conjunto de pessoas que ao longo deste quatro anos ajudaram a humanizar o caminho percorrido. Sem qualquer ordem especial, um profundo agradecimento a Liliana Ferreira, Catarina Oliveira, Sara Garcia, Paula Martins, Marlene Amorim, Mário Rodrigues e Liliana Silva. Grato pela vossa atenção, paciência, carinho e amizade.

E aos meus pais e irmã, com carinho, agradeço a atenção, o apoio, a paciência e muitas outras dádivas que me fizeram do fundo dos seus corações.

E aos companheiros caçadores de satélites: é tempo de se retomarem os trabalhos!

palavras-chave

Visualização interactiva, ventrículo esquerdo, segmentação de imagem, angiografia coronária por TC, análise functional

resumo

A angiografia coronária por TC (angio-TC) é prática clínica corrente para a avaliação de doença coronária. Alguns estudos mostram que é também possível utilizar o exame de angio-TC para avaliar a função do ventrículo esquerdo (VE). A função ventricular esquerda (FVE) é normalmente avaliada considerando as fases de fim de sístole e de fim de diástole, apesar de a angio-TC proporcionar dados relativos a diferentes fases distribuídas ao longo do ciclo cardíaco. Estes dados não considerados, devido à sua complexidade e à falta de ferramentas apropriadas para o efeito, têm ainda de ser explorados para que se perceba se possibilitam uma melhor compreensão da FVE. Para além disso, podem ser calculados diferentes parâmetros para caracterizar a FVE e, enquanto alguns são bem conhecidos dos médicos, outros requerem ainda uma avaliação do seu valor clínico.

No âmbito de uma utilização alargada dos dados proporcionados pelos angio-TC, este trabalho apresenta contributos ao nível da segmentação do VE e da sua análise funcional.

É proposto um método semi-automático para a segmentação do VE de forma a obter dados para as diferentes fases cardíacas presentes no exame de angio-TC. Foi também desenvolvida uma ferramenta de edição 3D que permite aos utilizadores a correcção das segmentações assim obtidas. Para a avaliação do método de segmentação apresentado foi proposta uma metodologia que permite a detecção de medidas de similaridade redundantes, a usar no âmbito da avaliação para comparação entre segmentações, para que tais medidas redundantes possam ser descartadas. A avaliação foi executada com a colaboração de três técnicos de radiologia experientes, tendo-se verificado uma baixa variabilidade intra- e inter-observador.

De forma a permitir explorar os dados segmentados, foram calculados vários parâmetros para caracterização global e regional da FVE, para as diversas fases cardíacas disponíveis. Os resultados assim obtidos são apresentados usando um conjunto de visualizações que permitem uma exploração visual sincronizada dos mesmos. O principal objectivo é proporcionar ao médico a exploração dos resultados obtidos para os diferentes parâmetros, de modo a que este tenha uma compreensão acrescida sobre o seu significado clínico, assim como sobre a correlação existente entre diferentes parâmetros e entre estes e o diagnóstico.

Finalmente, foi proposto um método interactivo para ajudar os médicos durante a avaliação da perfusão do miocárdio, que atribui automaticamente as lesões detectadas pelo médico ao respectivo segmento do miocárdio. Este novo método obteve uma boa receptividade e constitui não só uma melhoria em relação ao método tradicional mas é também um primeiro passo para a validação sistemática de medidas automáticas da perfusão do miocárdio.

keywords

Interactive visualization, left-ventricle, image segmentation, coronary CT angiography, functional analysis

abstract

Coronary CT angiography is widely used in clinical practice for the assessment of coronary artery disease. Several studies have shown that the same exam can also be used to assess left ventricle (LV) function. LV function is usually evaluated using just the data from end-systolic and end-diastolic phases even though coronary CT angiography (CTA) provides data concerning multiple cardiac phases, along the cardiac cycle. This unused wealth of data, mostly due to its complexity and the lack of proper tools, has still to be explored in order to assess if further insight is possible regarding regional LV functional analysis. Furthermore, different parameters can be computed to characterize LV function and while some are well known by clinicians others still need to be evaluated concerning their value in clinical scenarios.

The work presented in this thesis covers two steps towards extended use of CTA data: LV segmentation and functional analysis.

A new semi-automatic segmentation method is presented to obtain LV data for all cardiac phases available in a CTA exam and a 3D editing tool was designed to allow users to fine tune the segmentations. Regarding segmentation evaluation, a methodology is proposed in order to help choose the similarity metrics to be used to compare segmentations. This methodology allows the detection of redundant measures that can be discarded. The evaluation was performed with the help of three experienced radiographers yielding low intra- and inter-observer variability.

In order to allow exploring the segmented data, several parameters characterizing global and regional LV function are computed for the available cardiac phases. The data thus obtained is shown using a set of visualizations allowing synchronized visual exploration. The main purpose is to provide means for clinicians to explore the data and gather insight over their meaning, as well as their correlation with each other and with diagnosis outcomes.

Finally, an interactive method is proposed to help clinicians assess myocardial perfusion by providing automatic assignment of lesions, detected by clinicians, to a myocardial segment. This new approach has obtained positive feedback from clinicians and is not only an improvement over their current assessment method but also an important first step towards systematic validation of automatic myocardial perfusion assessment measures.

Contents

Contents i

List of Acronyms iv

1 Introduction 1
- 1.1 Motivation and Goals . . . 3
- 1.2 Main Contributions . . . 3
- 1.3 Published Work . . . 4
- 1.4 Outline . . . 5

I Context 7

2 Heart Anatomy and Physiology: A Brief Overview 9
- 2.1 The Human Heart . . . 11
- 2.2 AHA Conform Analysis of the Left Ventricle . . . 12
- 2.3 Heart Anatomy in CT . . . 14
- 2.4 Left Ventricle Function Analysis . . . 15
- 2.5 Conclusions . . . 16

3 Cardiac Computed Tomography 17
- 3.1 Historical perspective . . . 19
- 3.2 Multi Detector-Row CT: General Principles . . . 20
- 3.3 Heart Image Acquisition and Reconstruction . . . 22
- 3.4 Performing a Cardiac MDCT Study . . . 25
- 3.5 Conclusions . . . 28

4 Left Ventricle Segmentation and Analysis 31
- 4.1 Heart Segmentation Methods . . . 33
- 4.2 MDCT Validation . . . 40
- 4.3 Conclusion . . . 44

II Left Ventricle Segmentation 47

5 Left Ventricle Segmentation from Coronary CT Angiography 49
5.1 Introduction . . . 51
5.2 Exam and Image Characterization . . . 52
5.3 Image Pre-Processing . . . 52
5.4 Reference Phase Segmentation . . . 57
5.5 Full Exam Segmentation . . . 61
5.6 Prototyping . . . 63
5.7 Qualitative Evaluation . . . 66
5.8 CardioAnalyzer . . . 71
5.9 Conclusions . . . 71

6 Editing Tool for 3D Segmentations 73
6.1 Introduction . . . 75
6.2 3D Editing Tool . . . 76
6.3 Evaluation . . . 82
6.4 Voxel Editing and Binary Outline . . . 83
6.5 Discussion and Conclusions . . . 83

7 Similarity Measures for Left Ventricle Segmentations Comparison 87
7.1 Introduction . . . 89
7.2 Similarity and Discrepancy measures . . . 89
7.3 Preparatory Study to Choose Similarity measures . . . 94
7.4 Results and Discussion . . . 95
7.5 Conclusions . . . 101

8 Left Ventricle Segmentation Tool Evaluation 103
8.1 Introduction . . . 105
8.2 Background Work . . . 105
8.3 Experimental Protocol . . . 111
8.4 Evaluation Study Results . . . 114
8.5 Conclusions . . . 123

III Left Ventricle Analysis 125

9 Left-Ventricle Global and Regional Functional Analysis 127
9.1 Introduction . . . 129
9.2 Related Work . . . 130
9.3 Visualizations . . . 132
9.4 Left Ventricle Analysis . . . 137
9.5 Discussion and Conclusions . . . 143

10 Semi-Automatic Myocardial Perfusion Assessment 147
10.1 Introduction . . . 149
10.2 Current Myocardial Perfusion Assessment Protocol . . . 151

10.3 Interactive Tool for Myocardial Perfusion Assessment 152
10.4 User Feedback 162
10.5 Conclusions 163

IV Conclusions **165**

11 Conclusions and Further Work **167**
11.1 Overall Conclusions 169
11.2 Future Work 170

Appendices **171**

A Tools for Medical Image Processing, Visualization and Analysis **173**
A.1 Introduction 175
A.2 Toolkits 176
A.3 Software Applications 181
A.4 Discussion 190
A.5 Conclusion 195

B CardioAnalyzer **199**
B.1 Introduction 201
B.2 User Interface Conceptual Model 202
B.3 Main Features and Graphical User Interface 204
B.4 Conclusions 215

Bibliography **219**

Index **239**

List of Acronyms

AHA	American Heart Association
ASM	active-shape model
AUC	area under the curve
BCLV	biplane cine left ventriculography
CAD	coronary artery disease
CHVNG/E	Centro Hospitalar de Vila Nova de Gaia/Espinho
CMR	cardiac magnetic resonance
CT	computed tomography
CTA	coronary CT angiography
CTP	myocardial CT perfusion
DTMRI	diffusion tensor magnetic resonance imaging
EBCT	electron beam CT
ECG	electrocardiogram
ECHO	echocardiography
ED	end-diastole
EDV	end-diastole volume
ES	end-systole
ESV	end-systole volume
EF	ejection fraction
GHT	generalized Hough transform
GUI	graphical user interface
ICA	invasive coronary angiograph
LA	left atrium
LV	left ventricle
LVM	left ventricular mass
MDCT	multiple detector-row computed tomography
MI	myocardial infarction
MM	myocardial mass
MPR	multi-planar reconstruction
MPS	myocardial perfusion scintigraphy
MSCT	multi-slice computed tomography
MR	magnetic resonance
MRI	magnetic resonance imaging
QCA	quantitative coronary angiography
RA	right atrium
ROC	receiver operating characteristic
RV	right ventricle
RVEDV	right ventricle end-diastole volume
RVEF	right ventricle ejection fraction
RVESV	right ventricle end-systole volume

SPECT	single-positron emission computed tomography
SV	stroke volume
TPD	total perfusion deficit
TPR	transmural perfusion ratio
TTE	transthoracic echocardiography
VOI	volume-of-interest

Chapter 1

Introduction

"My purpose is to tell of bodies which have been transformed into shapes of a different kind."

– Ovid, The Metamorphoses

Coronary computed tomography (CT) angiography is now routinely used in clinical scenarios to assess the coronary arteries. Apart from the one or two cardiac phases typically used for coronary and left ventricle (LV) function assessment, the exam includes 10 or more cardiac volumes evenly spaced along one cardiac cycle. This large and usually neglected amount of data might provide additional insight over LV function. To explore such possibility several steps are required going from LV segmentation to a tool allowing visual exploration of the available analysis data.

This chapter presents the ideas that motivated the work carried out, outlines its main contributions and presents the structure of this thesis.

1.1 Motivation and Goals

Left ventricle (LV) function analysis is of paramount importance for cardiac assessment. With multiple detector-row computed tomography (MDCT) scanners, 4D ($3D + t$) cardiac exams, typically including 10+ cardiac volumes distributed along one cardiac cycle, can be performed. One of the major applications is coronary angiography to assess the coronaries (e,g., Wink et al. [1]). Several studies (e.g., Wu et al. [2] and Mahnken et al. [3]) have shown that MDCT exams also allow computing several left ventricle (LV) functional parameters (e.g., ejection fraction) which compare to those obtained using well established image modalities for cardiac analysis, such as magnetic resonance imaging (MRI) or echocardiography (ECHO).

Having several cardiac phases available (besides end-systole and end-diastole), it is possible to explore how different parameters vary along the cardiac cycle, thus allowing a more complete analysis and providing a chance to explore new parameters and analysis techniques. For this, it is necessary to segment the relevant structures for all the cardiac phases while dealing with a large amount of data (around 1.5 gigabytes per exam).

Segmenting a large number of cardiac phases can be a tiresome task. Even if the segmentation is performed automatically, there is always a need to revise/edit it to ensure its correctness. Tools such as CardioViz3D [4], which allow analysing cardiac data, are not suited for 4D analysis since they do not allow carrying it out as an integrated process, using knowledge from previously segmented cardiac phases to improve current phase segmentation and minimize required user interaction. One must not forget that, as important as the segmentation methods, are the tools which allow user interaction to guide the method or allow correcting the results [5].

Therefore, we proposed to provide a framework allowing the segmentation of the LV for multiple cardiac phases. Then, after obtaining such data we aimed to explore how LV function analysis could be improved by looking into different parameters characterizing LV function over time and how the clinicians could be supported in their analysis tasks concerning recent applications of CT cardiac angiography images, e.g., to perform myocardial perfusion assessment.

1.2 Main Contributions

The work presented in this thesis has the following notable contributions:

1. Left-ventricle segmentation method, from MDCT coronary angiography exams applied to all cardiac phases available (typically 10+, evenly distributed along the cardiac cycle);

2. Tool for 3D editing of left-ventricle segmentations allowing faster editing of the segmentations obtained using the proposed segmentation method;

3. Methodology proposed to support the choice of similarity/discrepancy metrics for left-ventricle segmentation comparison;

4. Left-ventricle segmentation evaluation methodology using global and regional based analysis;

5. Framework for the assessment of left-ventricle regional and global functional analysis based on simultaneous synchronized visualization of multiple parameters describing LV function;

6. Interactive tool to support visual myocardial perfusion assessment with automatic assignment of lesions to myocardial segments.

1.3 Published Work

The work presented in this thesis resulted in the following publications:

International Journals:

1. Samuel Silva, Beatriz Sousa Santos, Joaquim Madeira, "Exploring Different Parameters to Assess Left Ventricle Global and Regional Functional Analysis from Coronary CT Angiography", Computer Graphics Forum, accepted for publication, 2012

International Conferences:

1. Samuel Silva, Joaquim Madeira, Beatriz Sousa Santos, "Inter-observer Variability Assessment of a Left Ventricle Segmentation Tool Applied to 4D MDCT Images of the Heart", Proc. 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society – EMBC 2011, pp. 3411–3414, Boston, USA, September, 2011

2. Samuel Silva, Beatriz Sousa Santos, Joaquim Madeira, "Left-Ventricle Global and Regional Functional Analysis from MDCT Images", Proc. Ibero-American Symposium on Computer Graphics (SIACG11), pp. 217–223, Faro, Portugal, June, 2011 (2^{nd} best paper award and invited submission to Computer Graphics Forum)

3. Samuel Silva, Nuno Bettencourt, Daniel Leite, João Rocha, Mónica Carvalho, Joaquim Madeira, Beatriz Sousa Santos, "Myocardial Perfusion Analysis from Adenosine-Induced Stress MDCT", Proc. 5th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA11), LNCS 6669, pp. 717–725, Las Palmas de Gran Canaria, Spain, June, 2011

4. Samuel Silva, Beatriz Sousa Santos, Carlos Ferreira, Joaquim Madeira, Augusto Silva, "A preparatory study to choose similarity metrics for left-ventricle segmentations comparison", Proc. SPIE Medical Imaging 2011: Computer-Aided Diagnosis, vol. 7963, pp. 796326 – 796334, Orlando, Florida, USA, Feb., 2011

5. Samuel Silva, Joaquim Madeira, Beatriz Sousa Santos, Augusto Silva, "CardioAnalyser: A Software Tool for Segmentation and Analysis of the Left Ventricle from 4D MDCT Images of the Heart", Proc. 7th International Conference BioMedical Visualization (MediVis10, pp. 629–634, London, United Kingdom, July, 2010

6. Samuel Silva, Beatriz Sousa Santos, Joaquim Madeira, Augusto Silva; "A 3D Tool for Left Ventricle Segmentation Editing", Proc. 7th International Conference on Image Analysis and Recognition (ICIAR 2010), LNCS-6112, pp. 79–88, Póvoa do Varzim, Portugal, June, 2010

7. Samuel Silva, Beatriz Sousa Santos, Joaquim Madeira, Augusto Silva, "Processing, Visualization and Analysis of Medical Images of the Heart: An Example of Fast Prototyping using MeVisLab", Proc. 6th International Conference on BioMedical Visualization (MediVis09), pp. 165–170, Barcelona, Spain, July, 2009

8. Samuel Silva, Beatriz Sousa Santos, Joaquim Madeira, Augusto Silva, "Left Ventricle Segmentation from Heart MDCT", Proc. 4th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA09), LNCS 5524, pp. 306–313, Póvoa do Varzim, Portugal, June, 2009

1.4 Outline

This thesis is divided in five main parts with the contents organized as follows:

- **Part I**, **Context**, briefly covers fundamental concepts and notable work described in the literature concerning the context in which our work has been carried out:
  - **Chapter 2, Heart Anatomy and Physiology: A Brief Overview**, presents a brief overview concerning heart's anatomy and physiology along with a set of standards used for heart visualization and analysis.
  - **Chapter 3, Cardiac Computed Tomography**, presents the MDCT scanners used to obtain the cardiac images along with the main methods and protocols used for acquisition.

  - **Chapter 4, Left Ventricle Segmentation and Analysis**, concerns an overview of recent LV segmentation methods presented in the literature depicting a wide variety of approaches.

- **Part II**, **Left Ventricle Segmentation**, concerns the work developed to obtain left-ventricle data from CT cardiac angiography images:
  - **Chapter 5, Left Ventricle Segmentation from Coronary CT Angiography**, presents a LV segmentation method for multiple cardiac phases along the cardiac cycle.
  - **Chapter 6, Editing Tool for 3D Segmentations**, presents an editing tool for correcting the segmentations, if needed. Its main distinguishing feature is that it is a 3D tool.
  - **Chapter 7, Similarity Measures for Left Ventricle Segmentations Comparison**, presents a methodology to help choose similarity metrics for segmentation comparison. Starting from a wide range of metrics described in the literature it identifies metrics with similar performances and narrows the number of metrics which need to be used for comparison removing those which are redundant.
  - **Chapter 8, Left Ventricle Segmentation Tool Evaluation**, presents details regarding the evaluation of the proposed segmentation tool including intra- and inter-observer variability assessment from a set of segmentations performed by three radiographers.

- **Part III**, **Left Ventricle Analysis**, contains details regarding LV analysis:
  - **Chapter 9, Left-Ventricle Global and Regional Functional Analysis**, presents a framework for analysis of different parameters characterizing LV function showing how these vary along the cardiac cycle.
  - **Chapter 10, Semi-Automatic Myocardial Perfusion Assessment**, presents an interactive tool to support visual myocardial perfusion assessment.

- **Part IV**, **Conclusions**, presents overall conclusions of the work carried out and proposes some ideas for future work.

This document ends with two appendices: the first presents an overview on **Tools for Medical Image Processing, Visualization and Analysis** which helped choose the different software tools used for development; the second, **CardioAnalyzer**, presents the software application developed to support the work carried out.

Part I

Context

Chapter 2

Heart Anatomy and Physiology

A Brief Overview

"— I always tear up when the Grinch's heart grows three sizes.
— Tears seem appropriate. Enlargement of the heart muscle, or hypertrophic cardiomyopathy, is a serious disease which can lead to congestive heart failure."

– Sheldon, The Big Bang Theory, CBS

THE heart is one of the most important organs in the human body and an anatomically (and physiologically) complex structure. Along this text, there is a constant need to refer to different heart related concepts which, even though they could be explained *in situ*, we thought better gathered in a short chapter to which the reader could refer to, when needed. What follows is, therefore, not intended to be a thorough source of information about the heart.

2.1 The Human Heart

The main function of the heart (anatomy presented in figure 2.1) is to pump oxygenated blood to the whole body and de-oxygenated blood to the lungs. De-oxygenated blood enters the heart through the right atrium (superior and inferior vena cava), passes the tricuspid valve and is pumped to the lungs by the right ventricle, through the pulmonary arteries. Artery is the name given to any blood vessel which takes blood leaving the heart, while vein is a blood vessel taking blood into the heart. The oxygenated blood coming from the lungs, through the pulmonary veins, enters the left atrium, passes the mitral valve and is pumped to the entire body, by the left ventricle, through the aorta.

Given that the right ventricle has only to pump blood to the lungs and the left ventricle has to pump blood to the whole body, the latter has a thicker wall.

The papillary muscles can be found inside both ventricles and connect the mitral and tricuspid valves to the wall of the heart. These muscles contract in order to prevent blood from going through the valves during ventricle contraction.

The heart's wall is composed of three different layers. The inner layer, in contact with blood, is called the endocardium. The middle layer consists of muscle, responsible for heart contraction, is the myocardium. Finally, the outer layer is known as the epicardium.

The myocardium receives oxygenated blood through the coronary arteries.

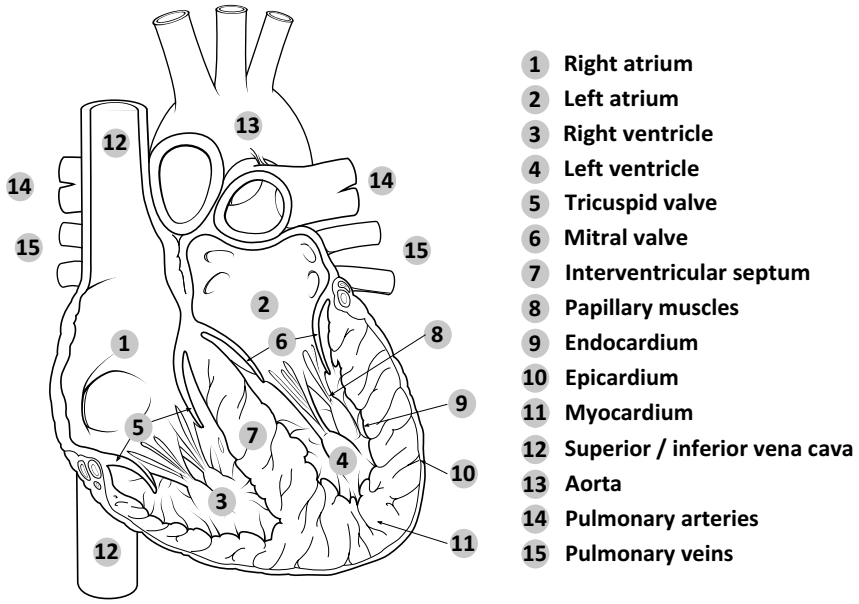


Figure adapted from source image obtained from MedScape,
Copyright © 2000 Healtheon/WebMD Corporation
http://images.medscape.com/pi/features/ald/cardio/cht-x.pdf/.

Figure 2.1 – Anatomy of the human heart.

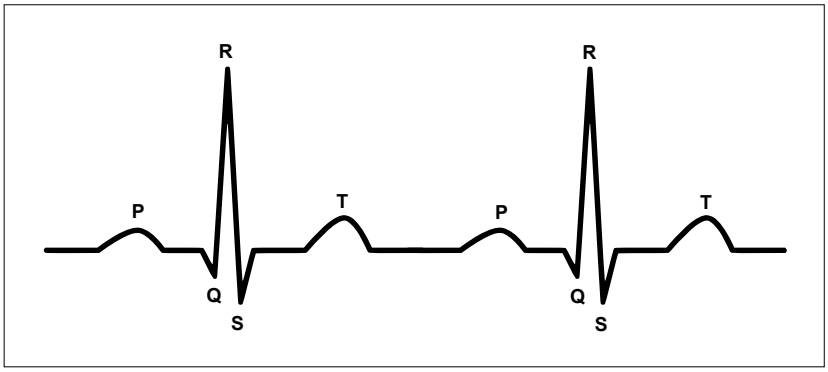


Figure adapted from source image obtained from WikiCommons,
http://commons.wikimedia.org/wiki/File:ECG-P%2BQRSkomplex%2BT.svg.

Figure 2.2 – Typical tracing of a normal ECG depicting some notable features.

Two important periods characterize the cardiac cycle: the diastole and the systole. The diastole occurs when the ventricles are passively filling with blood. When they reach their maximum volume of blood it is called end-diastole. The systole occurs when the ventricles contract. When they reach their maximum contraction (minimum blood volume) it is called end-systole.

The electrocardiogram (ECG) is an exam performed to measure the electrical activity of the heart. This is performed non-invasively by using several electrodes applied to the skin in the thorax and upper and lower limbs.

Figure 2.2 shows a typical tracing of a normal ECG. Five notable features of the ECG tracing are usually considered, denoted by the letters P, Q, R, S and T. Each of these features has a very precise meaning. For example, the P wave concerns atria depolarization. For the scope of the work presented in this thesis it is only important to note that it is common to refer to the cardiac cycle as the *RR interval*, i.e., the interval between two R peaks. Later on, when referring to percentages of the cardiac cycle these concern percentages of this interval.

2.2 AHA Conform Analysis of the Left Ventricle

The American Heart Association (AHA) has proposed [6] a set of standards to be followed for myocardial segmentation and a nomenclature to be used for all cardiac image modalities [7]. These standards are presented in what follows.

2.2.1 Cardiac View planes

Typically, CT images are viewed using three orthogonal planes. These planes are positioned parallel and at 90 degrees to the body long axis. These planes are presented in figure 2.3 and are

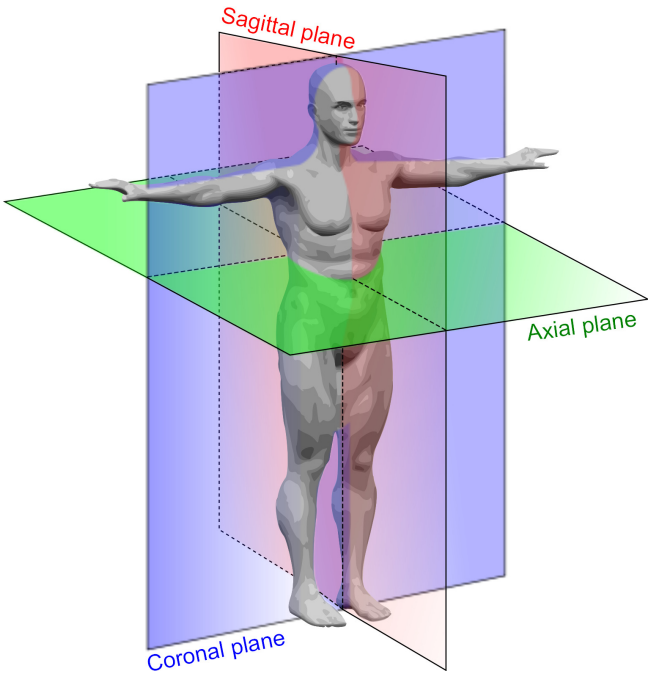


Figure adapted from source image obtained from WikiCommons,
http://commons.wikimedia.org/wiki/File:Human_anatomy_planes.svg

Figure 2.3 – Orthogonal planes typically used for CT image visualization and analysis: axial, sagittal and coronal.

named axial (cross-section to top-bottom axis), sagittal (cross-section to the left-right axis) and coronal (cross-section to the front-back axis).

The standard orthogonal planes do not provide the proper views for cardiac analysis since they fail to comply with the views traditionally used for 2D echocardiography which present the heart based on LV long axis instead of the body long axis.

Figure 2.4 shows the three planes commonly used to perform heart visualization and analysis: horizontal long axis, showing the two ventricles and atria; vertical long axis, showing the left ventricle and atrium; and short axis, perpendicular to the long axis, and showing both ventricles.

2.2.2 Myocardial Segments

As recommended by the AHA, the myocardium is usually divided in 17 segments used for analysis [6], by dividing the myocardium in three annular tomographic regions along its long axis (basal, mid-cavity and apical) and then dividing each of those regions in six, six and four angular segments respectively. A segment is included for the apical cap (not considered if the analysis is performed for the endocardium, i.e., the blood chamber). Each of these segments has a direct correspondence with the different regions of the polar map (also known as bull's-eye diagram)

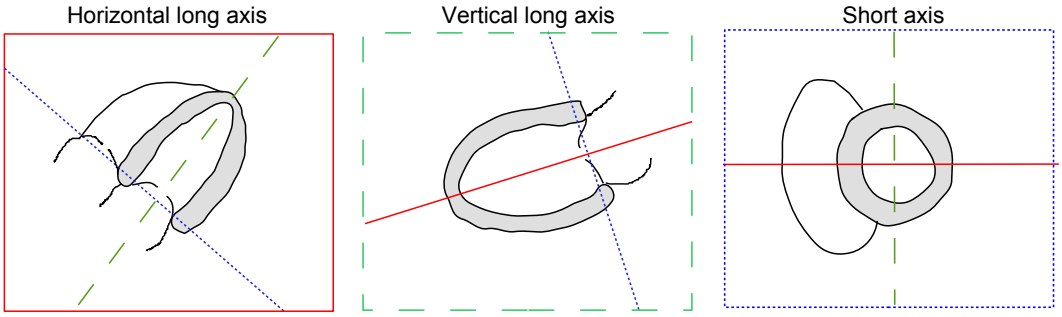


Figure 2.4 – Standard cardiac planes, as proposed by the AHA: horizontal long axis, vertical long axis and short axis.

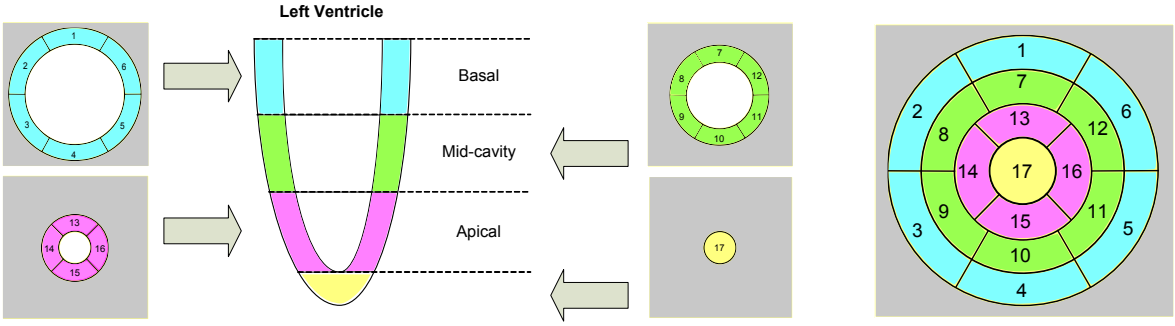


Figure 2.5 – Segments considered by clinicians when analysing the left ventricle and their placement in the polar map [6].

which is often used to depict myocardium analysis data (see figure 2.5).

This myocardial segmentation can be used, for example, to support the assessment of myocardium thickness or myocardial perfusion.

2.3 Heart Anatomy in CT

This section provides a brief overview regarding heart anatomy, considered relevant for the subject matter of this thesis, as it is shown in MDCT exams. For additional information concerning these exams and associated technology and methods the reader is forwarded to chapter 3.

Figure 2.6 shows the different cardiac view planes for a typical MDCT exam. The horizontal long axis view shows the LV, right ventricle (RV), left atrium (LA) and right atrium (RA). The orientation of the horizontal long axis plane might be changed, rotating it around the long axis, to provide a view including the outflow tract. This is usually known as a 5-chamber view since it shows both atria, both ventricles and the outflow tract. The vertical long axis view shows the LV and the LA.

Finally, the short axis view shows both LV and RV, in this particular case, close to the mitral valve level.

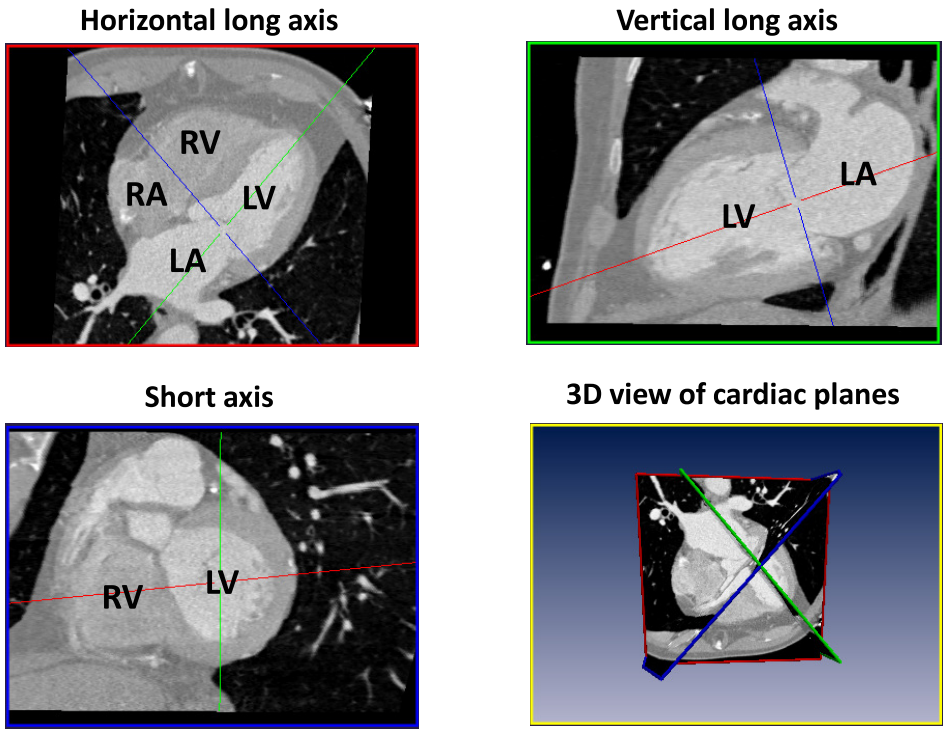


Figure 2.6 – Heart anatomy as seen on a cardiac CT using the standard cardiac planes as proposed by AHA.

2.4 Left Ventricle Function Analysis

The assessment of LV function is usually performed by computing several parameters [8], such as: stroke volume (SV) and ejection fraction (EF). These are computed based on the end-diastole (ED) and end-systole (ES) blood volumes. The SV is the difference between the end-diastole volume (EDV) and the end-systole volume (ESV). The EF is the ratio between the SV and the EDV and is given by:

$$EF(\%) = \frac{SV}{EDV} \times 100 = \frac{(EDV - ESV)}{EDV} \times 100$$

For MDCT exams, blood volumes are typically computed using one of two methods:

- Simpson's Method – This method consists in choosing several slices (N) along the LV long axis, computing the area for the blood pool in each (A_i) and then multiplying by section thickness (S_i). If cross-sections are obtained at regular intervals (fixed thickness, S), then the volume is given by:

$$Vol = \sum_{i=1}^{N} A_i \times S$$

- Voxel Counting – By identifying all voxels belonging to the blood pool and by considering image resolution (s_x, s_y, s_z) it is possible to compute the blood volume by multiplying the number of voxels, N_{voxels}, by voxel volume, V_{voxel} given by $s_x \times s_y \times s_z$:

$$Vol = N_{voxels} \times V_{voxel}$$

2.5 Conclusions

This chapter provided a brief introduction on the anatomy and physiology of the human heart. Its purpose was not to provide a thorough coverage of every aspect of the heart anatomy and physiology but just those deemed important for improved understanding of the following chapters.

Chapter 3

Cardiac Computed Tomography

"Any sufficiently advanced technology is indistinguishable from magic."

– Arthur C. Clarke

Cardiac exams, performed using multiple detector-row computed tomography (MDCT) scanners have a central role in the work carried out. Performing such exams follows a well established protocol which influences (and sometimes limits) the characteristics (quality) of the acquired images. Several factors should be taken into consideration, such as the amount of patient radiation dose, and image acquisition protocols are sometimes chosen to minimize these factors.

In what follows we provide a brief overview concerning how MDCT scanners work and how coronary CT angiography (CTA) is performed.

3.1 Historical perspective

The computed tomography (CT) technology has been in constant evolution [9] since the introduction of the first CT scanners in 1972 which earned Godfrey Hounsfield and Allan Cormack the Nobel Prizes for Medicine and Physiology in 1979.

The first scanners used a pencil-beam X-ray source and one or two detectors to acquire the data while the source was translated linearly across the field of view. After this the system was rotated by successive 1° steps until it covered a span of 180° (the maximum possible), which took longer than 5 minutes. The obtained image was 80×80 pixels.

This first approach was then improved by using a narrow fan-beam X-ray source and more detectors per row and then by introducing a wide fan-beam source and rotation of both source and detectors. The first cardiac imaging using CT was first reported in 1981, with a narrow fan-beam system and a gantry rotation time of 2 seconds which provided a temporal resolution between 500 and 1000 ms which was insufficient to obtain motion-free images of the heart.

In 1982 electron beam CT (EBCT) scanners were introduced. In these scanners both source and detector are stationary. Electrons are accelerated, precisely focused and swept across a 210° tungsten ring placed under the patient. On hitting the tungsten ring a cone-beam of X-ray photons is emitted, goes through the patient and is captured by two 240° detector rows above the patient. A major drawback of these systems is its high cost of purchase and maintenance.

In the late 1980's CT technology using spiral (or helical) acquisition appeared. In these systems the X-ray source and detectors are mounted on a sliding ring which allows a fast rotation of the gantry. The table, where the patient is lying, moves continuously along the longitudinal axis which allows covering the patient body in an helical course. With gantry rotation times of around 750 ms it was possible to perform volumetric acquisition of an anatomical region in 30 seconds with slices 2–10 mm thick which was still not adequate for heart imaging.

In 1993 multiple detector-row computed tomography (MDCT) was introduced. The existence of multiple rows of detectors reduces the time of examination. In 1998, a system with 4 rows of parallel detectors provided gantry rotation times of 500 ms and, using complex segmentation algorithm, reconstructions with temporal resolutions around 125 ms. The spatial resolution also got better with slice thicknesses of 1–1.25 mm.

The increasing number of detector rows (up to 16) allowed, in 2002, with gantry rotation times below 500 ms, obtaining cardiac volumes with isotropic spatial resolution, i.e., identical voxel size in the three planes (0.5 – 0.625 mm). Temporal resolution was also increased to values similar to those obtained with EBCT systems (53–65 ms).

These 16 row detectors, although a great advance, still required breath-hold times between 20 and 30 sec and for that reason systems have continued their evolution including 32, 40 and

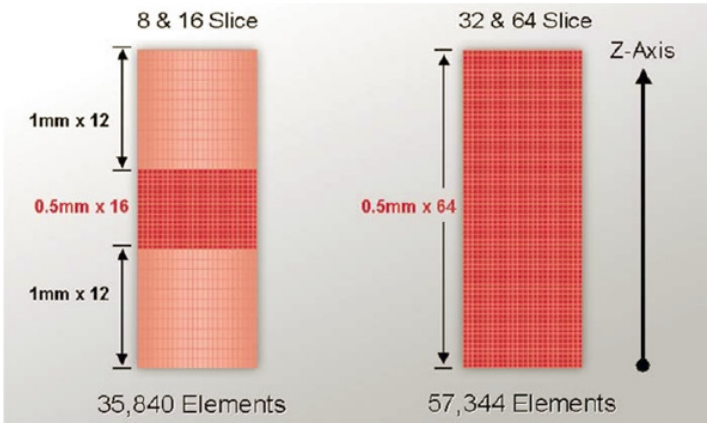


Figure 3.1 – Detector arrangement in adaptive (left) and fixed (right) arrays. Image obtained from [11].

64 detector rows which reduced scanning time considerably and allowed breath-hold times lower than 10 seconds.

Nowadays there are already systems being tested with 320 detector rows and spatial resolutions reaching 0.2 mm.

3.2 Multi Detector-Row CT: General Principles

This section presents some details regarding MDCT scanners and how they work, focusing on some of the most important parts, concepts and scanning techniques.

3.2.1 Detectors Geometry

Detectors geometry has been changing at a fast pace with an increasing number of rows but also aiming to provide the technician with multiple options regarding spatial resolution, exam time and consequently radiation dose limitation [10].

There are two main types of detectors: fixed and adaptive array. In fixed array detectors, the elements are identical in size, while in the adaptive array the detectors located in the centre are smaller than those on the outer side (figure 3.1).

An adaptive array allows an easier selection of the correct slice thickness for each examination providing a way of trading between spatial resolution and acquisition time. Even though current scanners have 64 or more detectors (current technology is reaching 320 detectors), which makes the acquisition time smaller, high resolution scans may suffer of a lower signal to noise ratio. Figure 3.2 shows how an adaptive array can be used to acquire 16 slices at different thicknesses.

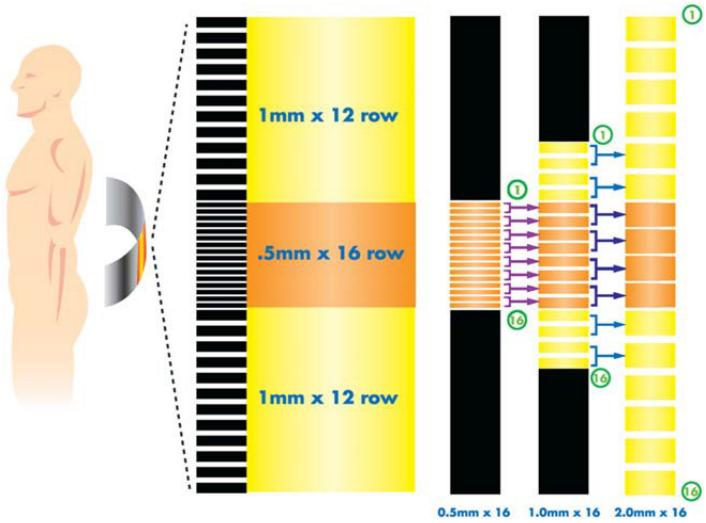


Figure 3.2 – Adaptive detector array with 16 0.5 mm elements in the center and 12 1.0 mm elements on each side. By using elements individually or by combining them it is possible to obtain several slice thicknesses. Image obtained from [10].

3.2.2 Rotation Time

Gantry rotation time has been rapidly increasing through the years (with typical current values around 400–500 ms), which allowed better temporal resolutions. Improved temporal resolutions lead to shorter examination times which is important, for example, to reduce the influence of patient motion on the acquired images. Even though it would be desirable to obtain faster rotation times there are physical limitations which prevent it. Due to the gantry rotation the X-ray tube suffers gravitational forces (g-forces) which increase exponentially with rotation frequency. As an example, a gantry rotation time of 500 ms implies 13Gs and a decrease to 400 ms an extra 7Gs [10]. Nevertheless, temporal resolution can still be enhanced by different reconstruction techniques described below.

3.2.3 Field of View (FOV)

MDCT imaging should aim (when possible) at acquiring data volumes with isotropic voxels. A single slice voxel has, in general, a greater size along the z-axis than in the x and y-axis. Even though this provides a good resolution in the axial plane it is not suited for multi-planar and 3D reconstructions because different data projections offer different resolutions [10].

The field of view (FOV), i.e., the region encompassed by the acquisition, affects voxel size and consequently (to maintain isotropic voxels), spatial resolution. With the standard CT imaging matrix having 512×512 pixels and a FOV of 25 cm, pixel size is approximately 0.5mm. To obtain isotropic voxels the resolution along the z-axis must also be 0.5 mm. On the other hand, if the

FOV is 50 cm then the plane pixel size increases to 1 mm and to maintain voxel isotropy, slice thickness must also be 1 mm. So, as a general principle, the FOV must be set to the minimum necessary to cover the anatomical region of interest and slice thickness adjusted accordingly.

3.2.4 Scanning Mode

Patient scanning can be performed in two different ways: sequential or helical (also known as spiral).

In sequential mode the patient table is stationary while acquisition is being performed. Then, the table moves to the next position and a new acquisition is performed, in general using electrocardiogram (ECG) triggering, i.e., the scan is always performed during a specific period of the patient's heart rate (further details are provided on Section 3.3.1). This scanning mode results in a radiation exposure which is far lower than that found in helical mode and is mostly used for coronary calcium scoring [7].

In helical mode, a scanner can simultaneously acquire data from all detector rows while the patient table is being moved. Helical acquisition can be characterized by its pitch, defined as the distance the table travels in a complete gantry rotation time divided by the size of the used detector array (number of slices $\times$ slice thickness, also known as collimation.) [7]. So, if a system has 64 detectors, with 0.625mm each, it results in a collimation of 40mm and to obtain a helical pitch of 1.0 the table can me moved at a rate of 40mm per gantry rotation. It is possible to attain higher table speeds while maintaining a helical pitch of 1.0 by using higher slice thicknesses thus trading resolution for speed. Basically, this means that a larger anatomic region can be covered at the cost of decreased detail and higher noise. A helical pitch lower than 1.0 means that the table moves a smaller distance than that covered by the detectors which results in slice overlap. In cardiac acquisitions it is common to use pitch values of 0.2 and 0.4. Smaller pitches are also used in some orthopaedic and neural imaging situations.

Helical pitches greater than 1.0 can also be used, but the resulting images tend to be noisier due to the resulting under sampling. Nevertheless, smaller scanning times and lower radiation doses are important in several situations being a pitch value between 1.2 and 1.4 a good compromise for most examinations.

3.3 Heart Image Acquisition and Reconstruction

Due to cardiac motion and to minimize its effects on the acquired images two main solutions exist: prospective ECG triggering and retrospective ECG gating. Both scanning methods also

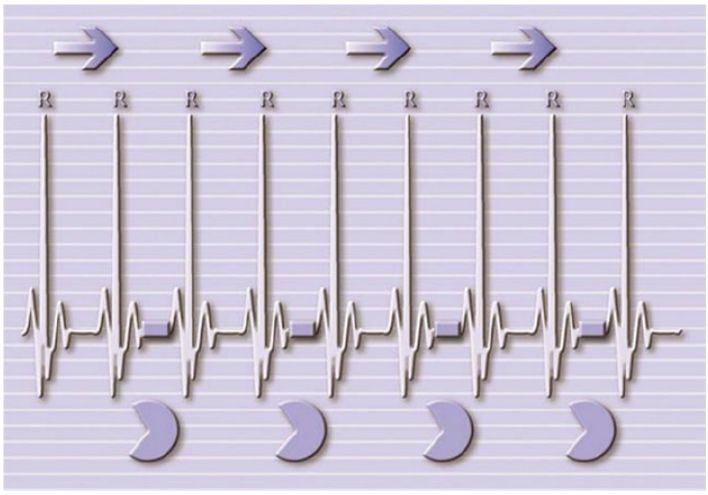


Figure 3.3 – Prospective scanning method, showing the acquisition window during diastole. Image obtained from [11].

have different impacts on patient radiation dose. For further information on this topic the reader is forwarded to [12, 13, 14].

In what follows a description of the main aspects of each method are presented.

3.3.1 ECG-triggered – Prospective Scanning

In prospective mode, the scanning is performed at a predetermined time window on the cardiac cycle, usually during diastole when there is less cardiac motion due to the fact that the ventricles are passively filling [9]. In this mode the ECG signal is used to control scanning: X-rays are only emitted at a predefined offset relative to the R-waves of the patient ECG signal, either given as a percentage of the RR-interval or in milliseconds [15].

The minimum data to allow image formation is obtained in a fraction of the complete rotation of the gantry (usually 240°–260°) [11]. The reconstruction algorithms used allow a temporal resolution equal to half the gantry rotation time (i.e., 200 – 320 ms).

The scanning is performed after discrete displacements of the table, which means that it must move the total collimation width after each acquisition. In general, due to the table displacement time one heart beat is skipped between acquisitions. Figure 3.3 illustrates this proceeding.

For each scan a number of slices equal to the number of detectors is acquired. The total time needed to obtain images of the whole heart depends on heart rate and number of detectors on the scanner. Scanners with 16 detector rows can cover the heart with a slice thickness of 1 mm in one breath hold [10].

Prospective scanning is usually used for coronary artery calcium scoring and as the X-ray

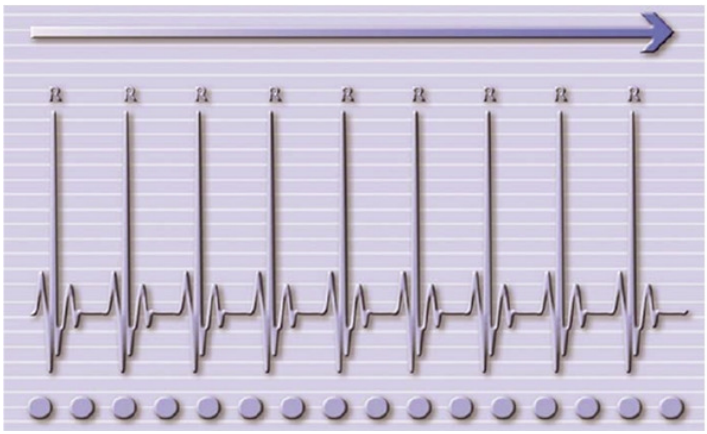


Figure 3.4 – Retrospective scanning method, showing the diagrams of the continuous helical acquisition. Image obtained from [11].

source only emits for short periods along the cardiac cycle it results in a low patient radiation dose.

One of the main issues which can influence prospective scanning is that of a high or irregular heart rate. For a heart rate of 60 bpm (beats per minute), with a RR interval of 1 second it is possible to acquire images in 200 ms with negligible motion artefacts. But, above 90 bpm, acquisition failures are sure to occur. Heart rate irregularity due to atrial fibrillation or other arrhythmias may also result in lower image quality.

3.3.2 ECG-gated – Retrospective Scanning

In retrospective mode the heart volume is covered continuously by a spiral scan while the ECG is recorded. In general, a typical examination may result in nine or more image volumes reconstructed at 10% intervals from a value of 5% RR interval up to 95% RR interval [10]. Figure 3.4 illustrates this mode.

It is necessary that the whole heart is covered by the exam during the cardiac cycle which implies that the table must not move more than the total width of the active detectors for each heartbeat: overlapping data sets are acquired with low helical beam pitch values. The helical pitch can be adjusted, according with the heart rate, to achieve continuous volume coverage. The typical pitch for an average heart rate of 70 bpm is 0.3:1, with a total scanning time around 20 seconds for a 16-slice MDCT scanner [9].

Retrospective scanning is usually used to evaluate the coronary arteries or veins, the thoracic aorta and the pulmonary veins. Due to the reduced size and tortuosity of the coronary vessels, their evaluation requires higher spatial resolution than that required for calcium scoring.

What results from this scanning mode is that the patient receives a higher dose of radiation. This can be reduced by ECG-gated dose modulation which consists in decreasing X-ray output in systole reducing radiation doses up to 48% [16].

Two main algorithmic approaches are used to perform image reconstruction based on retrospective cardiac gating: partial scanning and segmented adapted scanning. Both these techniques depend on the heart rate.

To reconstruct an image, a minimum helical projection data segment of 180° must be available or a temporal resolution of 200–250 ms (half-rotation time for a 16-slice MDCT scanner which is why this is also known as half scan reconstruction). As all the data must be acquired within a single cardiac cycle, higher heart rates can be an issue. This technique works well with heart rates up to approximately 70 bpm. Above that rate it is usual to administer beta-blockers to patients, reducing the heart rate and allowing better conditions for partial scanning.

To improve temporal resolution, segmented adaptive reconstruction can be used which involves combining shorter segments of projection data from two or more subsequent cycles [9]. This technique can be used in patients with a higher heart rate because the acquisition time is reduced (see figure 3.5).

Figure 3.6 shows an example of how using both reconstruction techniques over the same dataset can have very different results. Notice how the multi segment reconstruction allowed for a more reliable analysis due to better image quality.

For greater detail on the reconstruction approaches used in MDCT imaging scenarios the reader is forwarded to Flohr et al. [17].

3.4 Performing a Cardiac MDCT Study

A MDCT study is performed following a specific protocol [18]. The following sections give a brief overview of the main steps involved.

3.4.1 Patient Preparation

Given the necessity of holding breath for some seconds and having a suitable (low) heart rate, patients are submitted to a preliminary exam mainly to test their capability of holding their breath for the required amount of time and if there is any heart rate variability during that time. To control anxiety or high heart rate, patients may be given some specific medication such as mild sedatives or beta-blockers.

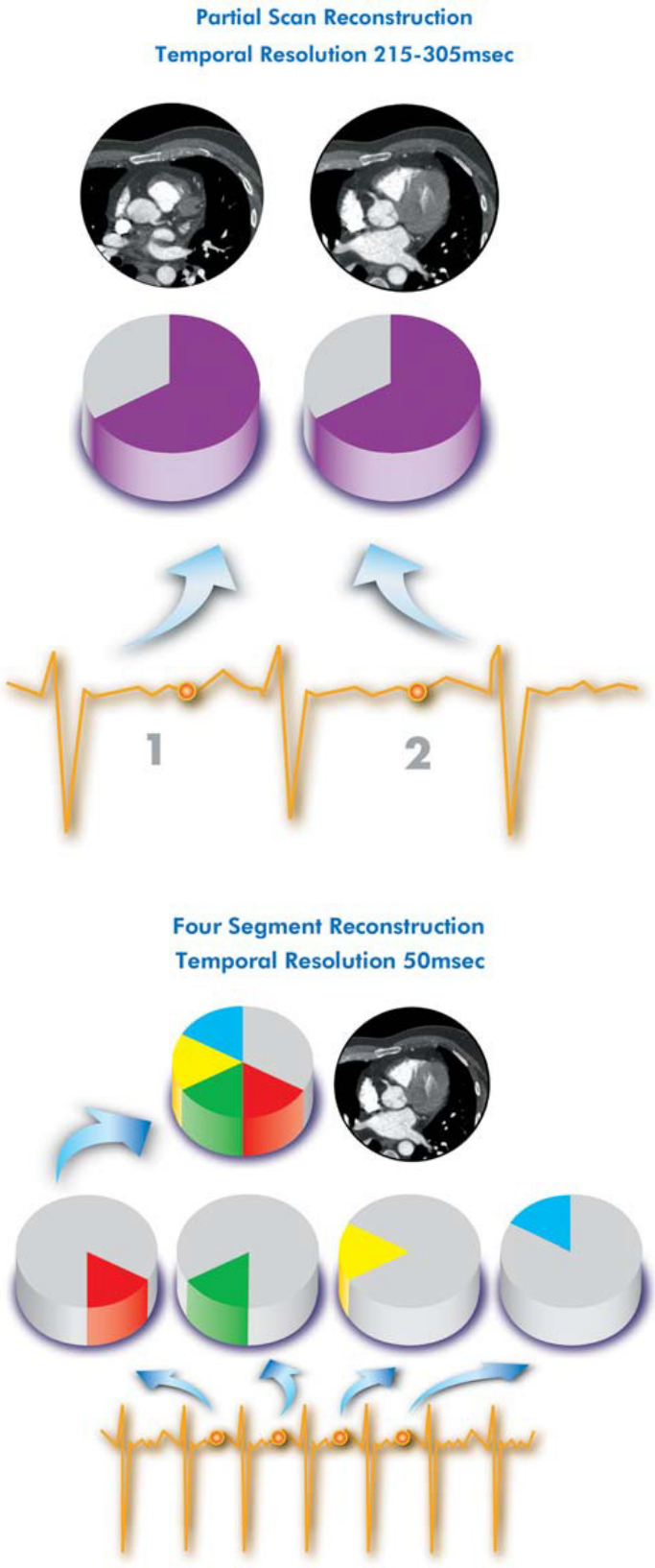


Figure 3.5 – Images depicting half scan reconstruction (top) and multi segment reconstruction (bottom) to improve temporal resolution. Image obtained from [10].

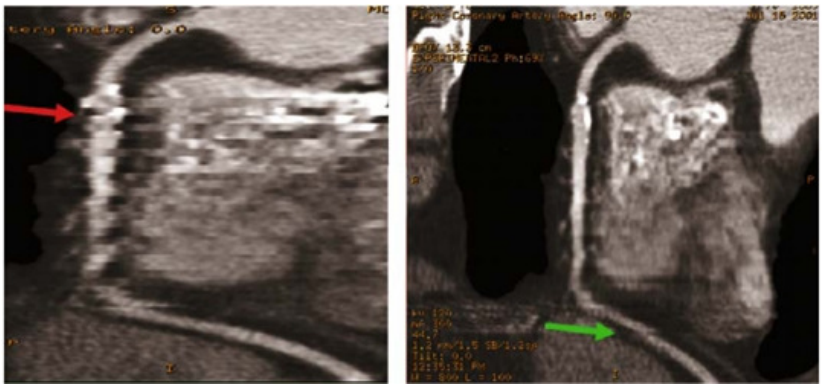


Figure 3.6 – Images obtained from the same dataset by half scan reconstruction (left) and multi segment reconstruction (right). Notice how the left image exhibits a large number of motion artifacts, making the analysis impossible, which almost completely disappear when using multi segment reconstruction. Image obtained from [7].

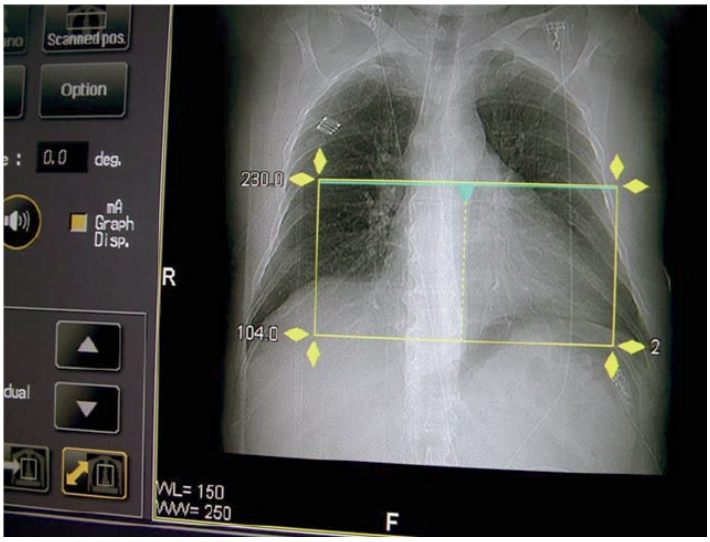


Figure 3.7 – Acquisition volume being defined with the help of a topogram. Image obtained from [11].

3.4.2 Image Acquisition and Contrast Administration

The imaging process starts by the definition of the acquisition limits. This is accomplished by performing a topogram (also know as scanogram or scout scan) (figure 3.7).

Based on the anatomical information obtained with this first scan and still without contrast administration a second scan is performed (using prospective scanning) to assess the presence and amount of calcium in the coronary arteries (figure 3.8).

The topogram is also used to define the region of interest for the coronary angiography study. This study implies the administration of a contrast agent to enhance the vascular tree and it

Figure 3.8 – Non contrast prospective scanning showing coronary arterial wall calcification. Image obtained from [11].

is important that the images are obtained when the contrast agent is on the arteries. This is accomplished by triggering the acquisition with the arrival of the contrast to the aortic root using a technique of bolus tracking: a continuous scan is performed (one axial slice) at the aortic level and the Hounsfield units are measured. When a specific threshold is reached (around 150 HU), which conveys an adequate opacification of the aortic root the acquisition of the whole heart volume is started.

3.4.3 Image Analysis

After acquiring the data, volumes are reconstructed using several techniques, choosing the most adequate set of images for coronary artery visualization and rejecting those which may introduce artefacts due to heart or patient movement [11].

The retrospective reconstruction of the heart is then searched focusing on approximately 60% of the RR interval. The ECG is also analysed looking for irregularities (e.g., premature beats) which might have introduced artefacts on the images. Current systems provide tools such as automatic filters to minimize these issues.

The information obtained from the retrospective reconstruction, which is not used for the coronary arteries analysis, is still useful as it can be used to perform a functional analysis, e.g., of the ejection fraction [11].

3.5 Conclusions

This chapter provides a short overview regarding MDCT scanners, its main features and the typical image acquisition protocol concerning coronary CT angiography (CTA).

Three important ideas should be kept from this chapter. First, the heart is a moving organ. Therefore, cardiac acquisition is not a simple task and is supported by different technologies and acquisition/reconstruction methods which make it possible, but with inherent limitations such as time resolution. Second, patient radiation dose reduction is a major concern. Therefore, every possible method is used to pursue that goal. As a consequence, image quality is not always the best (even considering limitations due to heart motion) and a considerable amount of noise and image artefacts can be observed, in particular around the end-systolic phase, when the heart is contracting and a low dose of radiation is delivered.

Finally, it is important to notice that MDCT technology is evolving at a fast pace (with increasing importance in clinical practice [19]) and some of the limitations might be overcome (e.g., concerning image quality) when recent 320–MDCTs become common. This fast evolution (including increasing spatial and time resolution) also results in advantages over other image modalities such as magnetic resonance imaging (MRI) namely regarding availability, exam time and its applicability, for example, to patients with pacemakers.

Chapter 4

Left Ventricle Segmentation and Analysis

> "Because learning does not consist only of knowing what we must or we can do, but also of knowing what we could do and perhaps should not do."
>
> –Umberto Eco, The Name of the Rose

During the past decades research regarding cardiac image analysis has been extremely active given not only its importance for healthcare but also due to a continuous development on image diagnosis equipment which provides an increasing (and improved) amount of data.

Different image modalities have been used for heart imaging such as magnetic resonance imaging (MRI) and multiple detector-row computed tomography (MDCT) to name just a few.

For diagnosis purposes the obtained cardiac images have to be analysed to extract relevant information. Computerized systems are now widely used to help on the process by providing tools for segmentation and analysis of the different heart structures allowing an improved exploration of the available image sets. Hence, during the last decades, researchers have been prolific in proposing several segmentation and analysis techniques to be applied to data resulting from different cardiac imaging modalities.

This chapter presents an overview on the methods used to segment the left ventricle from coronary angiography exams performed using MDCT followed by a short review on relevant literature concerning MDCT validation as a reliable source for left ventricle analysis.

4.1 Heart Segmentation Methods

Even though a vast amount of work has been presented concerning heart segmentation from multiple image modalities, with magnetic resonance imaging (MRI) taking the lead (see Kaus et al. [20], Lynch et al. [21], Berbari et al.[22], Mille et al. [23], Lee et al. [24] and Jolly et al. [25] for recent examples), this section will only focus on segmentation methods applied to multiple detector-row computed tomography (MDCT) images.

Recent surveys have been presented by Suri et al. [26] and Frangi et al. [27] which provide a very complete overview on what has been done in the past two decades[1] concerning left ventricle (LV) segmentation. Given the completeness of those surveys, the segmentation methods presented in this section are just some examples of recent work deemed important to characterize current methods and trends.

4.1.1 MDCT Based Segmentation Methods

The presented segmentation methods can be grouped into two main classes [1]: model-less and model-based[2]. A few examples of both classes are presented below.

Model-less methods

Model-less methods are characterized by dealing with the segmentation task using a pipeline including image processing methods such as threshold and region-growing and often requiring user intervention to initialize or guide the method (e.g., orienting the image according to specific rules or defining regions of interest). As a result some of these methods tend to be time consuming and the outputs might depend on how user intervention was performed (e.g., where seed points were placed).

Jolly et al. [30] present a segmentation method consisting on two main steps applied to manually reformatted image volumes into sort-axis slices: global localization and local deformation. On the global localization step the clinician segments the left ventricle for one basal slice in the end-diastole (ED) phase using thresholding and connected component analysis. The main goal is to identify the LV endocardium by searching for a large connected component that is least eccentric, most circular, and most convex and then approximate it by a circle. The epicardium is also approximated by a circle with the same centre but of larger radius. The system then propagates this segmentation to the remaining ED slices (ED propagation). The article is not clear regarding how this is performed since at one point it states the thresholding is repeated

[1] Higgins et al.[28] is a notable example of early work on the field.

[2] An alternative grouping has also been proposed by Ma et al. [29]: threshold-based, clustering-based and model-based.

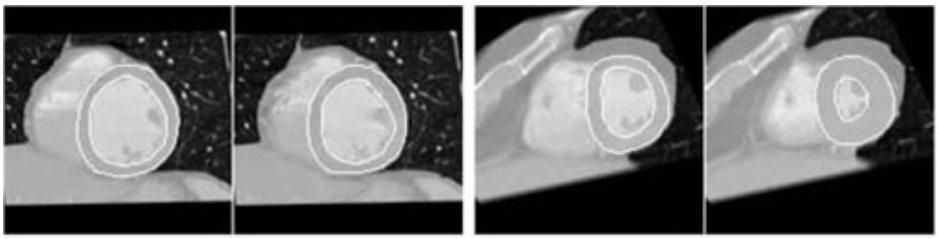


Figure 4.1 – Several examples of the obtained myocardium segmentation. Image obtained from [30].

for the remaining ED slices but further ahead contour scaling and copying is referred as the ED propagation method. To propagate these initial localization contours to the end-systole (ES) phase (ES propagation) scaling and copying is applied to the previously determined ED contours. Since the endocardium contracts more than the epicardium, the endocardium contour is scaled by 0.6 and the epicardium by 0.9. These global localizations can also be propagated along the remaining phases (time propagation) by copying the ED contours.

On the local deformation step the myocardium region is segmented using the Expectation-Maximization algorithm [31] and active contours. A shape constraint is introduced, for time and spatial propagation situations, by using contour data from the previous image as a template which is warped to the candidate data points obtained. Image 4.1 shows some example segmentations. It is important to note that this is the segmentation method present in the Siemens Argus[3] package, provided by one of the workstations available in Centro Hospitalar de Vila Nova de Gaia/Espinho (CHVNG/E).

Muhlenbruch et al. [32] present a semi-automated 3D region growing method to segment the LV endocardium. After an initialization, defining a short-axis plane at the base of the LV (mitral valve) and choosing a point inside the blood chamber, the method performs region growing inside the blood chamber (not including the papillary muscles). This segmentation method is provided by Siemens Circulation[4], one of the software packages available at CHVNG/E for LV segmentation. The reported time for segmenting the endocardium for two cardiac phases is around 170 seconds.

Fleureau et al. [33] [34] present a segmentation method based on what they call a multi-agent approach. Although it is a generic method, which can accomplish the simultaneous segmentation of different structures in a image, it is applied to the segmentation of the heart.

The method starts with user defined seed points (from where worker agents will depart) in each one of the structures to be segmented (see figure 4.2). Each of these agents will then try to conquer the pixels/voxels around it based on their similarity concerning local intensity and texture

[3]Siemens syngo Argus: `http://www.medical.siemens.com`.
[4]Siemens syngo Circulation: `http://www.medical.siemens.com`.

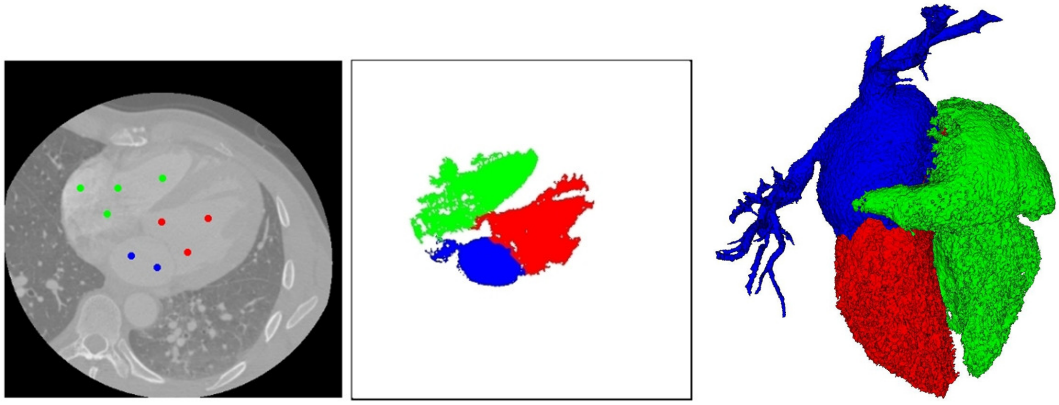


Figure 4.2 – On the left, initial seed points are set and labeled according to the structure they belong to (left ventricle (red), left atrium (blue) and right ventricle (green)). On the centre, the segmented regions and, on the right a 3D visualization of the segmentation result. Image obtained from [33].

of the region the seed point was put in. A controller agent manages the evolution of all worker agents. When a worker agent decides to conquer a pixel it asks permission to the controller agent which has three options:

- *The pixel/voxel has not been associated to any structure yet*: the controller marks it as occupied, computes the distance the worker had to travel to reach it (for future reference) and authorizes the worker to grab the pixel and continue;
- *The pixel/voxel is already associated with the structure the worker agent belongs to*: the controller computes the distance the worker had to travel to reach the pixel and compares it with the distance travelled by the worker agent that previously grabbed the pixel/voxel. Only the shortest distance is retained and the corresponding worker agent allowed to grab the pixel/voxel;
- *The pixel/voxel has already been assigned to a different structure*: as in the previous case, the distance travelled by the current worker to reach the pixel/voxel is computed and compared with the distance travelled by the worker agent which grabbed the pixel/voxel. The one travelling the shorter distance will be allowed to grab the pixel/voxel.

A 3D visualization of the obtained output is presented in figure 4.2. With this method it would be difficult to segment structures as the myocardium since its intensity and texture are similar to those found in other regions of the image. To cope with this kind of situation, the authors included inhibitor agents which block the worker agents progression. They basically work like normal worker agents which are associated with a region which does not belong to any

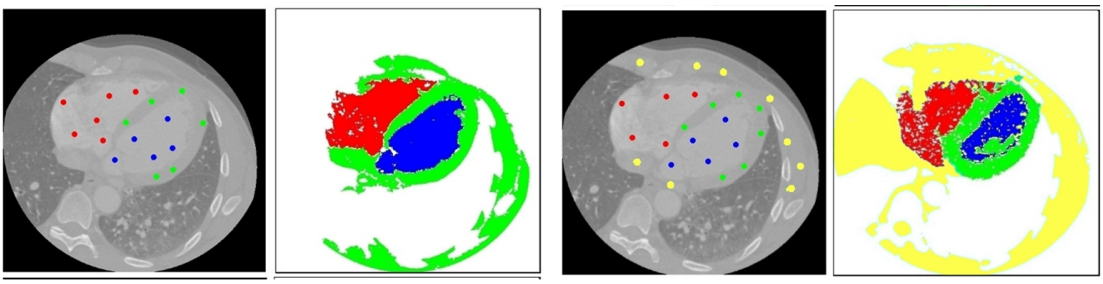


Figure 4.3 – On the left, seed points have been selected on the myocardium but the segmentation outflows to other regions of the heart with similar intensity and texture. On the right, by introducing inhibitor agents the myocardium is properly segmented. Image obtained from [33].

anatomical structure of interest. Figure 4.3 shows how the presence of these inhibitor agents can result in a correct segmentation of the myocardium. It is not clear how the number and position of the seed points can influence the segmentation results.

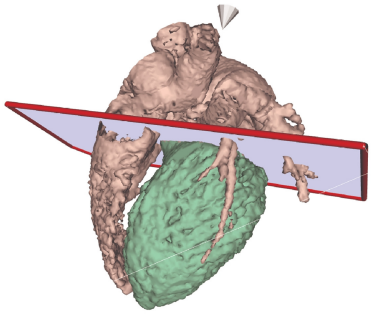


Figure 4.4 – Selecting the mitral valve plane to limit the LV segmentation (presented in different colour). Image obtained from [35].

Fleaureau et al. have also presented an improved version of this method by using a fuzzy connectedness approach to deal with the way the segmentation is carried out [35]. It basically improves the robustness and speed of classifying new pixels/voxels as part of a structure. This improvement was applied to multi-phase endocardium segmentation from MDCT images. Although the setting of seed points inside the LV and left atrium (LA) is mentioned, in order to avoid the segmentation progressing through the LA (through competition between agents), the mitral valve plane seems to be defined interactively using a cutting plane as can be seen in figure 4.4.

Okuyama et al. [36] present a 3D region growing method for LV segmentation. This segmentation method requires user initialization by setting a short-axis plane on the base of the LV, at mitral valve level, and a mouse click inside the lumen of the LV in each phase to be segmented (typically ES and ED, which have to be visually determined). After segmentation, the upper segmentation threshold can be adjusted in order to cope with small differences in contrast enhancement level, particularly close to the endocardium.

Model-based methods

Model-based segmentation methods approach the segmentation task based on a statistical model describing the most notable properties of the region to segment. This statistical model is created from a training set of previously performed segmentations (annotations) and is then adapted to fit each new image by varying model parameters. The general approach usually starts by a rough location estimation (which might include user intervention) followed by an adaptation of the model to the current data.

Metaxas et al. [37] use a new class of deformable models, metamorhps [38], to segment the LV and right ventricle (RV) endocardia and epicardia.

The main innovation regarding the general definition of metamorphs, which can be applied in a wide variety of scenarios, as shown in [38], is the presentation of a new automatic method to initialize the deformable models in order to speed up the convergence and minimize the chance of it being enclosed inside a local minimum. Based on an initial seed point (and its surroundings in a 5 pixel radius) the method computes intensity statistics which it uses to build an image intensity probability map. Based on this map it estimates a region of interest (ROI) which limits the boundaries to which the model will evolve. It then determines the skeleton for this ROI and uses its coordinates to compute a covariance matrix. An ellipsoid is then built with its axes oriented according to the matrix's eigenvectors and scaling according to the eigenvalues.

Figure 4.5 shows the initialization and model evolution for a LV image.

Van Assen et al. [39] present a LV segmentation method based on active-shape models (ASMs) [40] which uses fuzzy inference for edge detection during model matching. Fuzzy inference was used as an alternative to the classic method of using statistical grey-level models in each sample point to generate model updates since it eliminates the need of having to retrain the model (which means building new training sets) for different imaging modalities (e.g., MDCT, MRI, etc.).

The training data set consisted in 53 magnetic resonance (MR) exams which were segmented manually and the method was applied to both MR and MDCT images.

Before model matching the initial position, orientation and scale have to be manually defined setting the model to a size not larger than the actual LV in the image and desirably covering a large number of slices.

Ecabert et al. [41, 42] present a model-based, fully automatic segmentation of the whole heart including both ventricles (endo- and epicardial borders included for the LV), both atria, LV epicardium and great vessels (aorta, pulmonary artery and pulmonary veins).

In a first stage, a cardiac model was built. The model is composed by different separate meshes, one for each of the anatomical images to be segmented. Figure 4.6 shows the mean

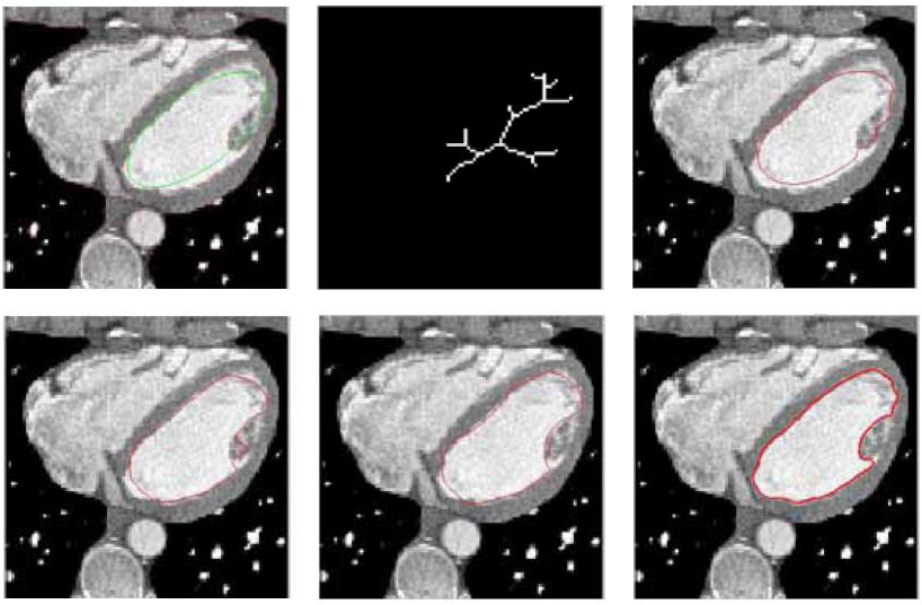


Figure 4.5 – Top row, from left to right: initial deformable model based on the ROI estimation, ROI skeleton and model after first iteration. Bottom row, from left to right: model after 10^{th} iteration, model after 50^{th} iteration and final result. Image obtained from [37].

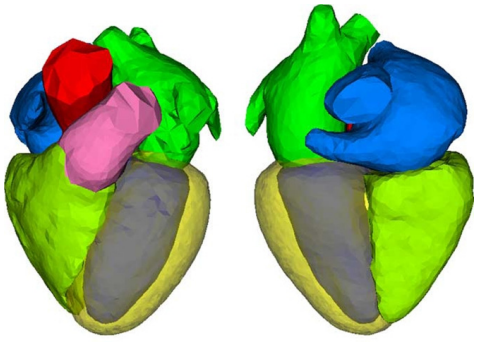


Figure 4.6 – Mean model composed by LV (endocardium in dark blue, epicardium (semitransparent yellow), LA (green), RV (light green), RA (blue), trunk of the aorta (red) and pulmonary artery (pink). Image obtained from [42].

model created.

The training data set consisted of 28 computed tomography (CT) image volumes corresponding to different cardiac phases from 13 patients.

The segmentation method has four main steps: heart localization, parametric adaptation using similarity transforms, parametric adaptation using affine transforms and deformable adaptation.

For heart localization a modified generalized Hough transform (GHT) [43] is proposed. The main aspect of this phase is that the GHT is not applied to the original image but to a preprocessed

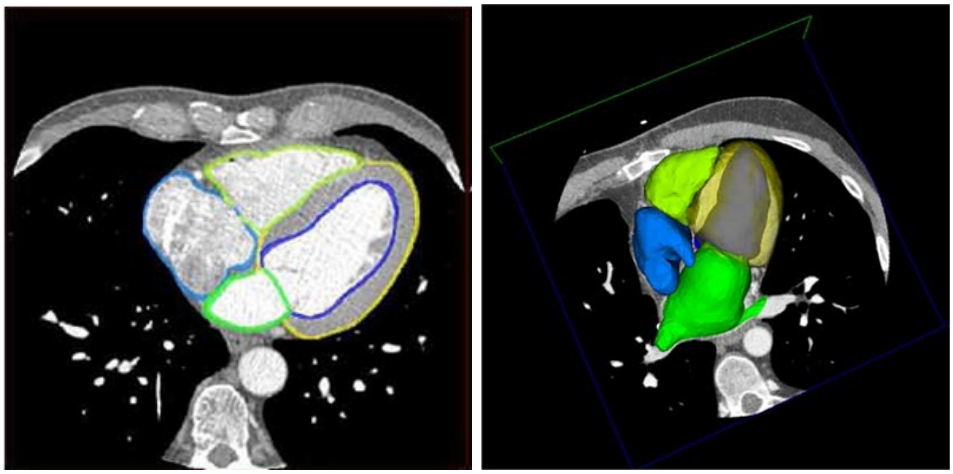


Figure 4.7 – Final segmentation of the four heart chambers depicted over a slice of the main image (left) and fully rendered (right). Image obtained from [42].

version resulting from a subsampling, thresholding and edge detection in order to eliminate some irrelevant structures surrounding the heart and reduce the amount of necessary memory. The subsampling is performed to obtain a voxel size of $3 \times 3 \times 3$ mm^3 which is still enough to preserve the heart structures while eliminating the small structures in the surrounding area.

The first step of parametric adaptation is used to globally position the heart model according to the image data. This is performed by applying a similarity transform (including translation, rotation and scaling) to adapt the model to the heart position, orientation and size.

The second parametric adaptation step will take care of resizing and deforming each element of the model to adapt it to the anatomy and cardiac cycle phase.

The last segmentation step concerns deformable adaptation which deals with the accurate adjustment of each element to the anatomical region it corresponds to.

Figure 4.7 shows an example for all these stages and the final result with the model superimposed to the image.

Zheng et al. [44] [45] also present a heart segmentation method based on ASMs. This method allows the segmentation of both endo- and epicardial borders of the LV with an identification of the mitral valve and including the outflow tract, left and right atria and right ventricle with explicit identification of the inflow and outflow tracts. The method consists in three main steps: model creation, 3D object localization and non-rigid deformation.

A mesh model is created for each chamber (see figure 4.8 for an example for the LV). The training set was composed of 323 cardiac CT volumes with the four chambers manually annotated plus 134 cardiac CT volumes with only the LV annotated. This larger number of exams with LV annotation is to improve the method's performance for the LV given its greater clinical importance

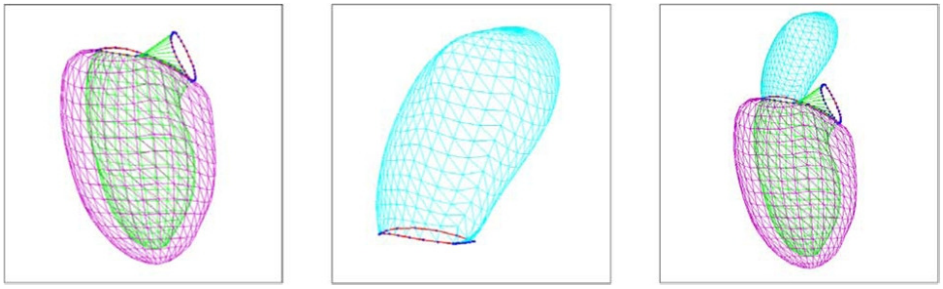


Figure 4.8 – Meshes created for the LV and LA. The LV epicardium in magenta, the LV endocardium in green and the LA in cyan. Image obtained from [44].

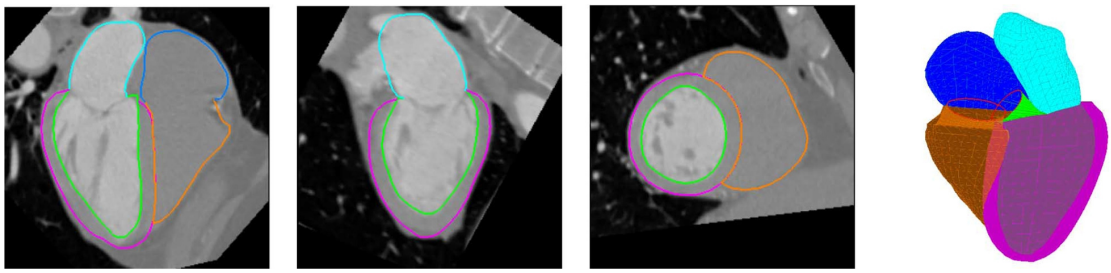


Figure 4.9 – Three orthogonal view planes and 3D mesh of four-chamber heart model: LV endocardium (green), LV epicardium (magenta), LA (cyan), RV (orange) and RA (blue). Image obtained from [44].

when compared with the other heart chambers.

In the second step of the method, 3D object localization is performed using marginal space learning and steerable features by training and applying estimators for position, position and orientation and similarity transform. Notice that the second estimator, for position and orientation, is not in any way redundant (considering the first estimator is already for position). The output of the first estimator is a set of position hypothesis which are then tested using multiple orientations by the second estimator which produces a different (narrower) set of hypothesis which are then processed by the third estimator (adding the scale variation).

It follows non-rigid deformation. After aligning the mean shape model for each chamber according to the pose determined earlier, it starts shape deformation to adjust the mesh to the chamber boundaries. Figure 4.9 shows an example of the final segmentation for all heart chambers.

4.2 MDCT Validation

Cardiac analysis using MDCT imaging, supported by segmentation tools such as those described above, has been increasingly used not only due to the constant developments concerning better spatial and time resolutions, but also due to a set of comparative studies which classify it as a

reliable tool for heart analysis.

Since MRI has been established as a gold standard (side-by-side with echocardiography) most of those comparisons have been performed using MR images as reference. To start, MDCT has, in general, a greater availability, lower costs and shorter exam times. On the other hand, image modalities such as MRI do not include the injection of a (nephrotoxic[5]) contrast agent nor any radiation.

At this point the question is "Can the data obtained using coronary CT angiography (CTA) for the study of the coronaries be additionally used for global and regional functional analysis of the heart?"

Schlosser et al. [46] compare analysis data obtained by manual and automatic segmentation of the left ventricle from 16-MDCT coronary angiography images and MR images analysed using a commercially available software (Argus, Siemens). An image reformat was applied to the MDCT images in order to obtain 8 mm thick slices in the short-axis plane as provided by MR.

A comparison is performed using computed data concerning end-diastole volume (EDV), end-systole volume (ESV), ejection fraction (EF) and left ventricular mass (LVM). According to the presented results, the EDV and ESV computed from the manual and automatically segmented data obtained from the MDCT images were higher than those obtained from MRI. The opposite happened with the LVM which was smaller for the MDCT based analysis. The EF was similar in the three analysis scenarios. The authors conclude that the overestimation of ESV and EDV might be due to MRI and MDCT not giving corresponding slices. A slight difference on the most basal slice defined can yield such differences. They therefore conclude that only EF can be reliably assessed from MDCT images.

Raman et al [47] compare heart function parameters computed from 16-MDCT cardiac angiography and MRI data (referred to as cardiac magnetic resonance (CMR) by the authors). The LV and RV have been segmented in both images using a semiautomated method provided by dedicated software (Mass Analysis v6.0 for MR exams and Advantage Windows 4.2 CardIQ III Function for MDCT exams, both from GE Healthcare). In order to have similar slice thicknesses for exams on both modalities the MDCT exams were reformatted in the short-axis plane to obtain thicker slices (8 mm, as provided by MR). Apart from the typical parameters characterizing LV function (EDV, ESV, EF and LVM), a comparison was also performed using RV analysis parameters (right ventricle end-diastole volume (RVEDV), right ventricle end-systole volume (RVESV) and right ventricle ejection fraction (RVEF)). A very good correlation has been found for all the analysed parameters between the two compared image modalities. A difficulty in tracing RV contours is reported by the authors given the poor contrast between RV blood pool and myocardium,

[5]Nephrotoxic: with poisonous effect on the kidneys.

motivated by the usage of a conventional coronary CT acquisition protocol (which is not optimized for right heart opacification). Wall motion analysis was possible in both image modalities but occasional low contrast or motion artefacts in the MDCT images presented some difficulties, revealing a need for additional advances in both acquisition speed and volume coverage for this modality.

Dewey et al. [48] also assess LV function using 16-MDCT (10 cardiac phases) and MRI comparing the obtained LV function parameters. Segmentation is performed using a semi-automated method provided by dedicated software (MR: ARGUS, Siemens AG; and MDCT: ANET, Toshiba). Once again, 8 mm thick slices have been generated from the MDCT data.

Good correlation has been found for all the compared parameters with just a slight overestimation of ESV and EDV which can be due to the limited temporal resolution provided by MDCT.

Chaosuwannkit et al. [49] compare the LV EF values obtained using 16-MDCT and biplane cine left ventriculography (BCLV). As happened with the previous comparison studies, the MDCT data is reformatted in order to obtain 8mm thick slices on the LV short-axis plane. Results show that the EF values obtained from both image modalities have a strong correlation. Nevertheless, the values for ESV and EDV are not provided which limits the comparison.

Butler et al. [50] compare LV functional heart analysis parameters obtained using 64-MDCT and 2D-transthoracic echocardiography (TTE) in patients with heart failure. All patients presented a low EF (below 45%) and different problems on the LV wall. Segmentation is performed using dedicated software (Circulation software, Siemens, Forchheim, Germany). Apart from the usual parameters (EF, EDV, ESV), regional motion assessment was also performed detecting multiple cases of dyskinesis, akynesis and hypokinesis. The authors found a good correlation between the data and diagnosis obtained from both image modalities.

Fischbach et al. [51] present a comparison between heart analysis performed using MDCT and MRI in patients with various cardiac diseases. When compared to MRI, the analysis performed using MDCT showed a good agreement for the EDV, ESV and a moderate agreement for the EF values with a tendency to slightly underestimate EDV and EF and overestimate ESV. The small difference observed in the EF is attributed to the lower temporal resolution of the MDCT which could not capture the exact ES phase. Regarding regional motion assessment, even though there was a good agreement between the two image modalities, MDCT based analysis tended to underestimate the motion problems.

Wu et al. [2] performed an extended study comparing 64-MDCT derived results with those obtained using echocardiography (ECHO), single-positron emission computed tomography (SPECT) and MRI. As in previous studies, the original MDCT images were reformatted in order to obtain

Table 4.1 – Main results obtained for the comparison of MDCT based functional analysis of the LV with analyses performed using other image modalities. **#S** – Number of subjects used for validation; **MP** – myocardial problems present; **#D** – Number of detectors present in the MDCT equipment; **Ref.** – Image modality used as reference.

Authors	#S	MP	#D	Ref. Eq.	EDV	ESV	EF	LVM	RWM
Schlosser et al [46]	18	?	16	MRI	△△	△△	•	▽▽	–
Raman et al. [47]	24	?	16	MRI	•	•	•	•	•
Dewey et al. [48]	33	?	16	MRI	△	△	•	•	–
Chaosu. et al. [49]	15	?	16	BCLV	–	–	•	–	–
Butler et al. [50]	25	*yes*	64	2D-TTE	•	•	•	–	•
Fischbach et al. [51]	30	*yes*	16	MRI	▽	△	▽	–	▽
Wu et al. [2]	63	*yes*	64	ECHO	••	••	••	–	–
	63	*yes*	64	SPECT	••	••	••	–	–
	63	*yes*	64	MRI	•	•	•	–	•
Mahnken et al. [3]	9	*yes*	16	MRI	•	•	•	–	•
	9	*yes*	16	SPECT	••	••	•	–	••
Schipper et al. [52]	57	?	64	SPECT	•	•	•	–	–
Sarwar et al. [53]	21	*yes*	64	MRI	•	•	•	•	•

• – Good Agreement •• – Better agreement with standard (MRI) than this modality
△/▽ – Slight overestimation/underestimation △△ / ▽▽ – Considerable disagreement

10 mm thick slices on the LV short-axis plane. One of the particularities of this study is that patient heart rate has not been controlled to suitable values (for MDCT exams). In fact, 43 out of the total 63 subjects which took part on this study had a heart rate above 70 bpm. At least in this study, the heart rate variability did not have significant effects on the obtained results.

The MDCT derived LV global analysis parameters showed a very good agreement with those obtained using MRI data. In fact, considering MRI results as a reference, MDCT presented better results than those provided by SPECT and ECHO. Concerning regional ventricular function (wall motion, with dyskinesis, akynesis and hipokinesis assessment) the comparison was performed only between MDCT and MRI showing very good agreement between the analysis performed with both image modalities.

Mahnken et al. [3] present a comparison between MRI, 16-MDCT and SPECT, concerning global LV function parameters and regional ventricular function (wall motion) in patients with a history of myocardial infarction (MI). MDCT and MRI images were manually segmented and as in the other studies the MDCT data was reformatted to obtain 8 mm thick slices on the LV short-axis plane.

Concerning global parameters, MDCT agreed well with MRI and considering the latter as a reference it provided better results than SPECT. There was also a good agreement between MDCT and MRI for regional wall motion, with MDCT outperforming SPECT (with MRI as a reference). Regarding the detection of perfusion abnormalities, MDCT showed good agreement with SPECT.

More recent studies concerning MDCT derived LV analysis validation have been presented such as those by Schipper et al. [52] and Sarwar et al. [53]. Given that the protocol followed in these studies is similar to those already described above, only the main findings are presented and can be found in table 4.1 along with a summary of the remaining comparison studies. It shows agreement information for several global LV function analysis parameters (EDV, ESV, EF and LVM), regional ventricular function (particularly regional wall motion, RWM) and a note on if the study included patients with myocardium problems (MP), e.g., myocardial infarction which might add some difficulty to the study regarding the quality of the obtained MDCT images. The agreement information is taken from the corresponding articles as stated by the authors. No cross comparison among the different studies has been performed regarding, for example, a comparison of correlation values.

Okwuosa et al. [54] present a study which, curiously, is not a validation of MDCT by another image modality, but the opposite, a sign that 64-MDCT imaging is already considered trustworthy as a standard in some applications. The authors goal is to validate LVM computation from SPECT using two different methods.

A survey presenting additional comparison studies (although most of them are older than those presented here) along with a discussion regarding the obtained results can be found in Juergens et al. [8] and Cury et al. [55].

MDCT based analysis and validation of the right ventricle has not be included. For notable examples see Aho et al. [56] and Müller et al. [57].

4.3 Conclusion

Several recent methods for LV segmentation from MDCT have been presented followed by a short overview concerning the validation of MDCT as a source of data for cardiac analysis.

Recently, a wide variety of methods have been presented for LV segmentation, ranging from more simple threshold-based methods to active shape models. The latter provide faster, more robust approaches to segmentation but require a more complex support framework. Notice, for example, the work of Zheng et al [45], where a training set of more than 350 manually performed LV segmentations has been used.

Developing a segmentation method should take into careful consideration the purpose of the method and the work that will have to be performed to implement it. Unfortunately, to the best of our knowledge, the methods described in the literature are not made available for general use and, notably, most of the best methods proposed can later be found in dedicated commercial workstations by Siemens or Shina.

Another important aspect to retain from this chapter is that at its current development stage (even considering just the most common technology available in hospitals, 64-MDCT) MDCT technology is already considered a good source of data for cardiac analysis when compared with ECHO and SPECT and increasing improvements in technology have a positive impact in derived analysis (e.g., Abbara et al [58], concerning EF).

Part II

Left Ventricle Segmentation

Chapter 5

Left Ventricle Segmentation from Coronary CT Angiography

"If you want to make an apple pie from scratch, you must first create the Universe."
– Carl Sagan

Several left-ventricle segmentation methods have been proposed in the literature. Nevertheless, to the best of our knowledge, none of those methods is readily available to obtain the necessary data to pursue our work concerning left-ventricle analysis methods. Therefore, as a first step of our work, a semi-automatic left-ventricle segmentation method for coronary CT angiography (CTA) images was developed. The work carried out included a prototyping stage using MeVisLab followed by the integration of the developed method in a framework based on the MITK library. A qualitative evaluation of the method, with the help of experienced radiographers, has also been performed to guide development.

A quantitative, more extensive, evaluation of the developed segmentation tool was also performed and further details are presented in chapter 8.

Publications:

The work presented in this chapter has been partially published in:

- Samuel Silva, Beatriz Sousa Santos, Joaquim Madeira, Augusto Silva, "Left Ventricle Segmentation from Heart MDCT", Proc. 4th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA09), LNCS 5524, pp. 306–313, Póvoa do Varzim, Portugal, June, 2009
- Samuel Silva, Beatriz Sousa Santos, Joaquim Madeira, Augusto Silva, "Processing, Visualization and Analysis of Medical Images of the Heart: An Example of Fast Prototyping using MeVisLab", Proc. 6th International Conference on BioMedical Visualization (MediVis09), pp. 165–170, Barcelona, Spain, July, 2009
- Samuel Silva, Joaquim Madeira, Beatriz Sousa Santos, Augusto Silva, "CardioAnalyser: A Software Tool for Segmentation and Analysis of the Left Ventricle from 4D MDCT Images of the Heart", Proc. 7th International Conference on BioMedical Visualization (MediVis10), pp. 629–634, London, United Kingdom, July, 2010

5.1 Introduction

As stated by Cury et al. [55], the full potential of multiple detector-row computed tomography (MDCT) technology is just beginning to be explored and further advances are essential, concerning not only new automatically computed parameters but also improved tools to enable further insight into the wide range of data now becoming available (e.g., for left ventricle wall motion). These new advances require specific data to be extracted from the exams and, therefore, segmentation methods are of the utmost importance.

Different approaches regarding left ventricle segmentation for a variety of exam modalities have been widely described in the literature in the past years (refer to chapter 4 for an overview) and range from 2D methods (e.g., Jolly et al. [30]) to more complex, model-based, approaches (e.g., Zheng et al. [44]). Unfortunately, to the best of our knowledge, no off-the-shelf segmentation method is available which could be used as a basis for our work. Even though segmentation tools are provided in the different workstations available at Centro Hospitalar de Vila Nova de Gaia/Espinho (CHVNG/E), none provides exporting the resulting segmentations. Therefore, we engaged in the development of a segmentation method which should:

- Be as automatic as possible but allowing user intervention;
- Segment both left ventricle endocardium and epicardium;
- Provide a segmentation for all cardiac phases included in the MDCT exam;
- Be validated by radiographers to ensure the reliability of the resulting data.

As previously stated (chapter 3) the different cardiac phases acquired have different image quality. To keep radiation dosage to a minimum, radiation is as low as possible while acquiring all cardiac phases except around the 60% phase. This is the cardiac phase typically used to perform coronary artery assessment and an additional amount of radiation is used to ensure better image quality. Furthermore, in this intermediate phase the heart has less movement and its shape varies slowly, while the ventricles are passively filling with blood, thus resulting in less image noise and acquisition artefacts. This results that the 60% phase has the best image quality among all available cardiac phases with a better distinction between the left ventricle (LV) walls and the rest of the heart.

Given the poor image quality of the remaining cardiac phases, and since we aim to provide a method for full exam segmentation, our approach starts by a segmentation of the 60% phase (henceforth referred to as *reference phase*) and then uses data from that phase to enhance the segmentation process for the remaining phases. Using a reference phase/segmentation to improve

further segmentations for different cardiac phases has also been proposed by Jolly et al. [30] but with a different approach to segmentation, based on contour delineation for one slice and its propagation along remaining slices and cardiac phases (as described in chapter 4).

The developed segmentation method can, therefore, be divided in three main stages: image pre-processing, reference phase segmentation and full exam segmentation.

This chapter starts by a brief characterization of the used images followed by an overview of the features pertaining each of the main segmentation stages, depicted in figure 5.1. It also presents how user intervention features have been added to allow tweaking the outputs of important steps in the devised segmentation method. It follows a brief description on how a first prototype was built using MeVisLab and how a simple qualitative study was performed to assess the quality of the provided segmentations. Guided by the results of this study the segmentation method was later integrated in a framework developed using the MITK library. The chapter ends with several conclusions drawn from the developed work.

5.2 Exam and Image Characterization

Images used for this work were obtained by multiple-detector computed tomography (MDCT) using a 64-Slice CT scanner (Somaton Sensation 64, Siemens Medical Solutions, Forchheim, Germany) with the following scan parameters: gantry rotation time of 330ms; tube voltage of 100kV; tube current of 500 – 700 mAs; electrocardiographic pulsing for decreasing radiation dose with full tube current applied at 55–65% of the cardiac cycle.

Each coronary CT angiography (CTA) exam contained data for at least ten cardiac phases, evenly distributed along the cardiac cycle. The image volume for each cardiac phase had a resolution of $512 \times 512 \times 256$. The resolution along the ZZ' axis might vary as it depends on the size of the volume of interest initially set by the radiographer.

All the exams were exported to DICOM[1] format and anonymized before being used.

5.3 Image Pre-Processing

Before LV segmentation two important steps need to be performed: noise attenuation and LV orientation estimation

[1]DICOM – Digital Imaging and Communications in Medicine: further information regarding the DICOM format can be obtained at `http://medical.nema.org/`.

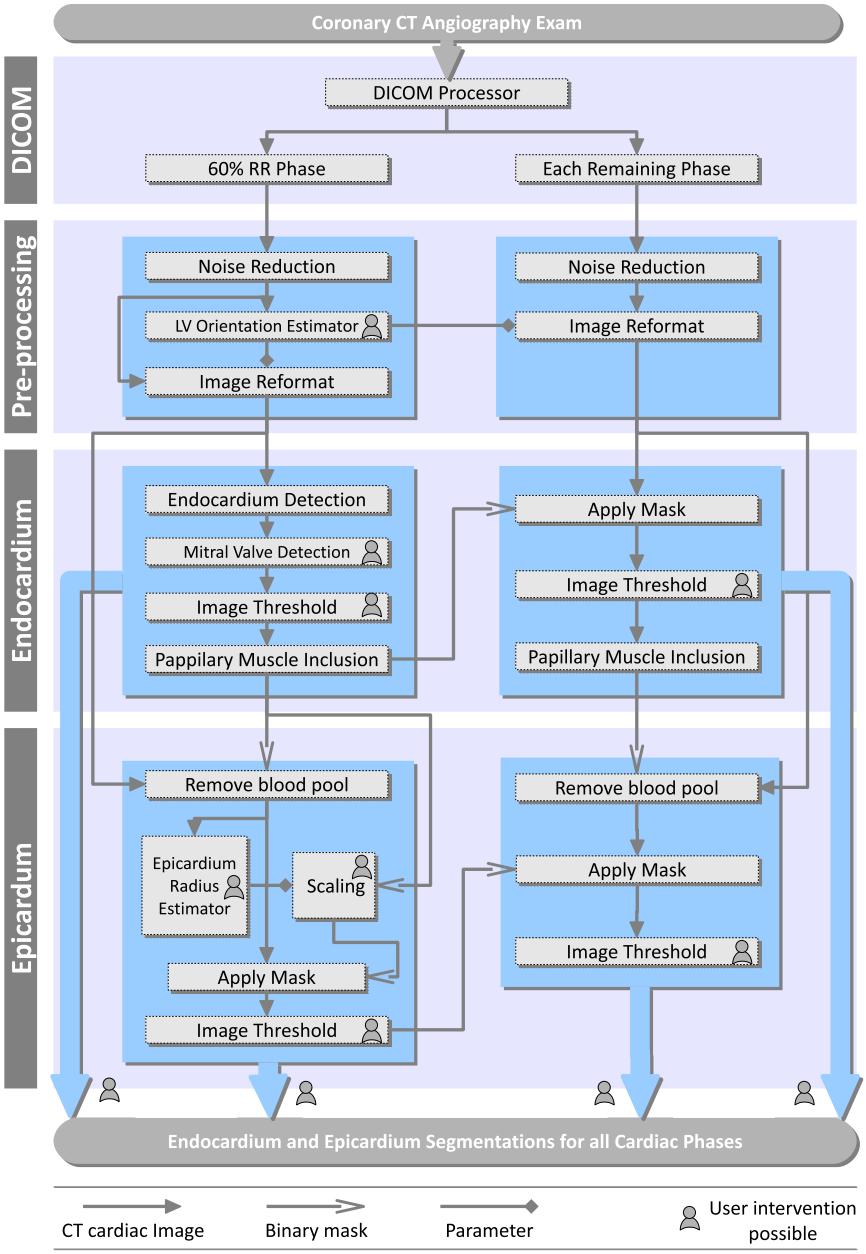


Figure 5.1 – Pipeline depicting the main steps for LV Segmentation for a full cardiac CT angiography exam.

5.3.1 Noise Reduction

Several methods have been described in the literature for image noise reduction (e.g., Pratt [59]). A few such methods were tested to improve image quality and only those are discussed bellow. All the methods were applied to CTA images and the results compared qualitatively by inspecting

the filtered images and testing overall segmentation method performance for each case.

A median filter is a non linear method for noise reduction [59]. For each voxel, its intensity is set to the median value of its neighbours. The size of the neighbourhood is defined by setting a radius (in number of voxels) along every dimension of the image. Even though the radius can be different in each dimension it is usually set to the same value resulting in a cube shaped neighbourhood. For example, setting a radius of 2 for all dimensions of a 3D image results in a $5 \times 5 \times 5$ neighbourhood.

Considering that our main interest is the LV and not any small sized structures, such as the coronaries, the median filter could be used with good results if the neighbourhood radius was kept small. In our case we managed to obtain the better results by setting neighbourhood radius to 1 and applying three iterations of the median filter.

Despite these good results we wanted to test if an edge-preserving method would result in any improvement.

Anisotropic diffusion is a method proposed by Perona et al.[60]. One of the main advantages of this method, when compared with other noise reduction methods, is its ability to preserve edges. Without entering into details regarding the mathematical definitions for anisotropic diffusion the important aspect to note is that an adjustable parameter, usually known as the conductance parameter, k, can be used to control the amount of smoothing applied to edges.

Anisotropic diffusion was applied to the images and various values were tested for the conductance parameter and the number of iterations. Conductance parameters between 5 and 8 and around 10 iterations provided good results and not much difference has been found between these results and those obtained using the median filter.

Therefore, given the similar results provided by the median and anisotropic diffusion filter and given the slightly higher computational cost of the latter we opted for the median filter. Nevertheless, the isotropic diffusion filter could have been used. In fact, an example of anisotropic diffusion applied to the same kind of images has been recently presented in Tsai et al. [61].

Figure 5.2 shows an example of the results obtained using the median filter. On the top row, the original image is presented showing the three standard cardiac planes (horizontal long-axis, vertical long-axis and short-axis). On the bottom row, the same image after median filtering.

5.3.2 Left Ventricle Axes Determination

Left ventricle axes must be defined in order to select a proper orientation for the data and segment the LV using slices along its long axis. It is particularly important to know LV orientation in order to perform segmentation up to (and parallel to) the mitral valve. Our first approach consisted in an automatic method to perform such task.

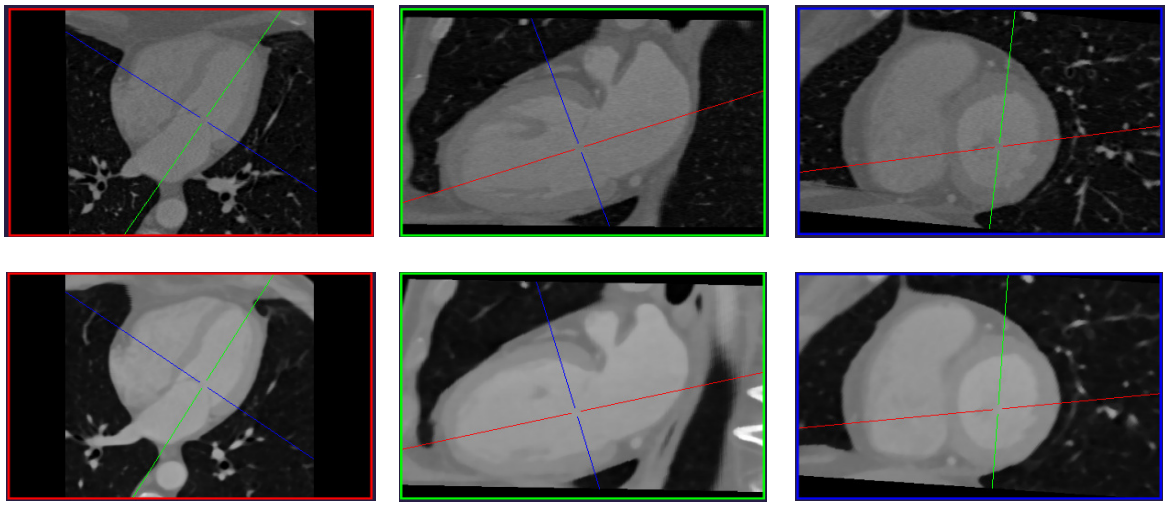


Figure 5.2 – Original image (top) and smoothed image (bottom) using a median filter.

To estimate LV position a coronal slice at mid-ventricular level is analysed. A threshold is performed and the image is searched from right to left looking for active regions. The first region found with a large area (to exclude regions related with the coronary arteries, lungs, etc.) is considered part of the left ventricle. Region growing is then applied in order to isolate it from all other active regions on the image (including, eventually, portions of the right ventricle where some contrast agent might be present). Finally, the centroid of the segmented region is computed and stored to be used as a reference during the segmentation of the remaining slices.

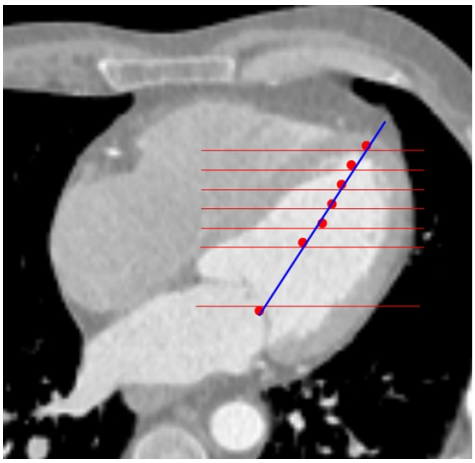


Figure 5.3 – Left ventricle axis determined from the centroids of the blood pool for different coronal slices.

In order to ease the detection of slices close to the left ventricle apex (where area cannot be used as a detection criteria), 3D region growing is applied to the entire set using the previously computed centroid as seed. This will isolate the left ventricle blood pool from other active regions related with the lungs and ribs, which might result in classifying a region as part of the left ventricle too early (ribs) or out of place (lungs).

The dataset resulting from the previously performed 3D region growing is then processed, slice by slice, using a coronal plane, from the left ventricle apex to its base. Each slice is searched for an active region from right to left. To ensure that none of the possibly remaining regions

are misclassified as left ventricle the one closest to the previously computed reference centroid is chosen. It follows region growing to segment that region.

To detect the stopping slice, the leftmost pixel of each segmented region is computed. If a sudden decrease (i.e., it moves to the left of the image) occurs it means that the region where the left ventricle connects with the aorta has been reached (outflow tract). From that moment on, our method looks for a sudden increase in the leftmost pixel of the segmented region which is related with the entrance in the left atrium region and segmentation stops.

Finally, the centroids for the first and last slices are used to compute the left ventricle principal axis.

This automatic method for LV orientation estimation provides reasonable results for most exams depending on the quality of the exam and anatomic characteristics of the LV.

Since this is an important step towards LV segmentation, users are allowed to supervise the obtained orientation. A "natural" way to do this is to, based on the computed data, set the orientation of the usual view planes used for cardiac analysis [6] (chapter 2) and ask users to approve or correct their orientation and position (as denoted in the diagram presented in figure 5.1).

To change plane location, i.e., where the planes cross, users just need to click with the mouse over the desired location. To change plane orientation users just need to click over the line representing the plane in any of the view windows and drag to attain the desired orientation.

While testing the LV orientation estimation method, users often performed corrections on the orientation and position of the different view planes, even if just to change orientation a few degrees tweaking the output of our method in order to reach the "optimum" setting. Observing this, we chose to test a different approach.

Since CTA is performed by setting an initial volume of interest (including the heart) the location and orientation of the LV are not completely different from patient to patient. Therefore, we estimated an average location and orientation from a set of patients and, instead of trying to automatically compute LV orientation, users are presented with this average orientation. In a case-by-case analysis this average orientation will typically be farther away from the "optimum" orientation than that proposed by our automatic method. Nevertheless, since changing plane orientation is quite simple, the work performed by users will not be much more than that performed to tweak our automatic approach. Furthermore, this second method avoids the computation time associated with orientation estimation (around 10s). Therefore, we kept using this second method. Nevertheless, the automatic method shows good potential and can be a good alternative if a fully automatic segmentation method is to be considered.

5.4 Reference Phase Segmentation

After obtaining LV orientation, image reformat is applied in order to perform the remaining segmentation steps using short axis planes. To perform myocardium segmentation we identify the region contained inside the endocardium, the region contained inside the epicardium and then subtract the latter from the first to obtain the myocardium segmentation.

5.4.1 Endocardium

To segment the endocardium two important steps are necessary: 1) locate the LV blood pool and 2) isolate it from the remaining heart and set the mitral valve plane, where the segmentation should stop (refer to figure 5.1).

When setting the view planes for cardiac analysis users typically position the crossing point close to the mitral valve level (see figure 5.4, top row). This provides important data because: 1) it clearly identifies where the LV is (the view planes crossing point (x_C, y_C) is inside the blood pool); 2) The mitral valve plane has been identified.

An alternative (and automatic) method for LV detection is to use a method similar to that described for orientation estimation but applied to the reformatted image.

With the data obtained from the view planes position and orientation, 3D region growing is applied to the image volume using (x_C, y_C) as seed. A low resolution segmentation is shown to the user ending at what is supposed to be the mitral valve plane (inferred from view plane position). Nevertheless, if the user positioned the view planes at mid-ventricular level (or inside the left atrium) it is possible to correct the segmentation by just using a slider, moving the current segmentation stopping plane up and down (see figure 5.4). The presentation of a low resolution segmentation is intended to provide smoother operation when interactively changing the mitral valve plane.

Finally, a high resolution of the segmentation is obtained. Since the papillary muscles are to be included as part of the blood pool and given that their corresponding attenuation level is lower and similar to that associated with the LV wall, to include those gaps a binary hole filling filter is applied to the previously segmented region.

The binary hole filling filter converts background voxels to foreground voxels based on the number of foreground voxels already existing in a pre-defined neighbourhood. The main idea is that if the number of foreground voxels is a majority (by a customizable amount), in the neighbourhood of a background voxel, that voxel becomes a foreground voxel. Three parameters control the performance of this filter: the neighbourhood size, the majority size and the number of iterations. The neighbourhood is defined by its radius in each dimension (e.g., a radius of

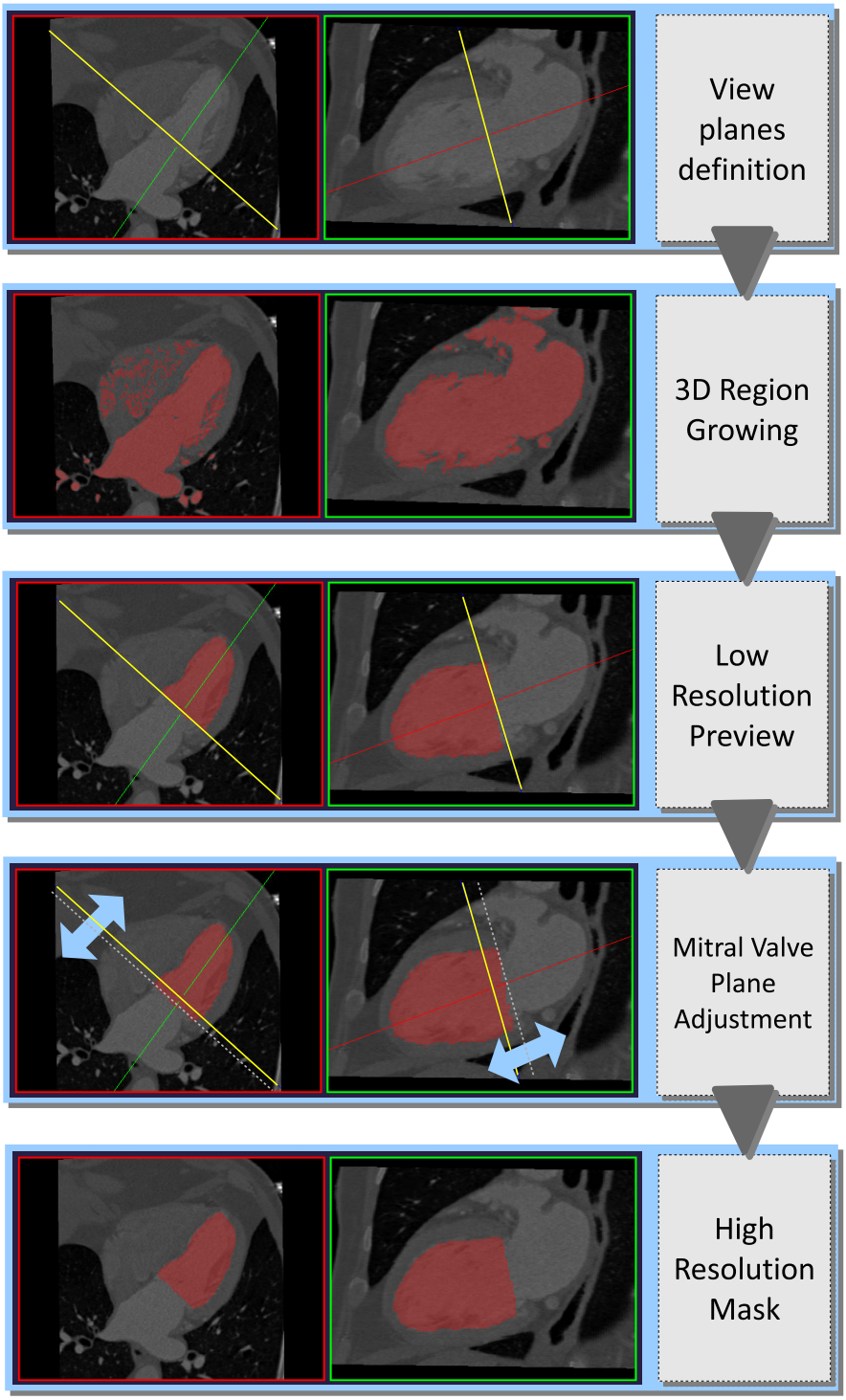


Figure 5.4 – Main steps for endocardium segmentation from the reference cardiac phase.

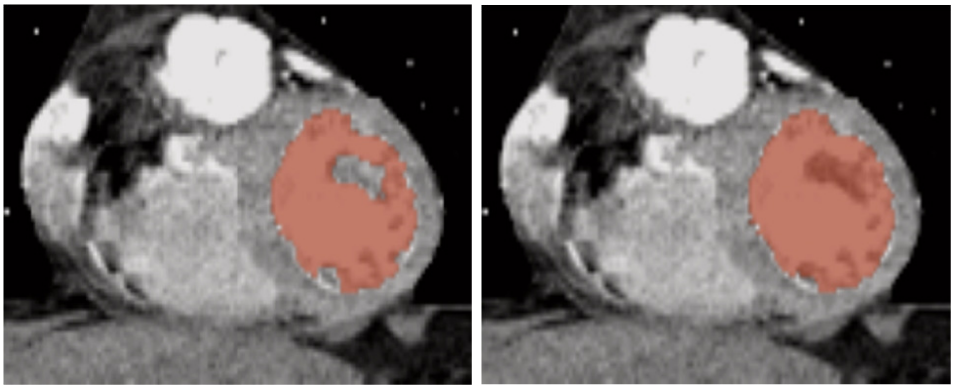


Figure 5.5 – Extracted blood pool before (left) and after (right) applying a hole filling filter to include the papillary muscles.

1 in all dimensions leads to a neighbourhood of $3 \times 3 \times 3$ voxels). The majority value is the minimum number of foreground voxels above half the number of neighbours needed to convert a background voxel into a foreground voxel (e.g., in a $3 \times 3 \times 3$ neighbourhood, the number of foreground voxels should be at least $(((3 \times 3 \times 3) - 1)/2 + majority\ value)$. Finally, the number of iterations defines the number of consecutive times the filter will be applied to the image.

Figure 5.5 shows the segmented region after the region growing filter (left) and after applying hole filling (right): the papillary muscles have been included in the blood pool.

The resulting segmentation can then be inspected by the user and edited using a 3D editing tool to add or remove regions to the segmentation. Additional details regarding the editing tool are provided in chapter 6.

Figure 5.4 presents the main steps for endocardium segmentation from the reference cardiac phase.

5.4.2 Epicardium

Since the papillary muscles are included in the blood pool and they present an attenuation range similar to that presented by the myocardium, they have to be excluded before epicardium segmentation. This is performed by removing the region segmented for the endocardium from the image volume (see figure 5.6).

The attenuation value range associated with the LV walls also appears in other (connected) regions of the heart and liver and given image quality it is often impossible to clearly distinguish where the myocardium ends and the inside of the right ventricle (RV) begins. So, to help segmentation, a region-of-interest (ROI) was automatically defined in order to remove most of the unwanted anatomical regions. This was performed in two steps (see figure 5.1): 1) Estimation

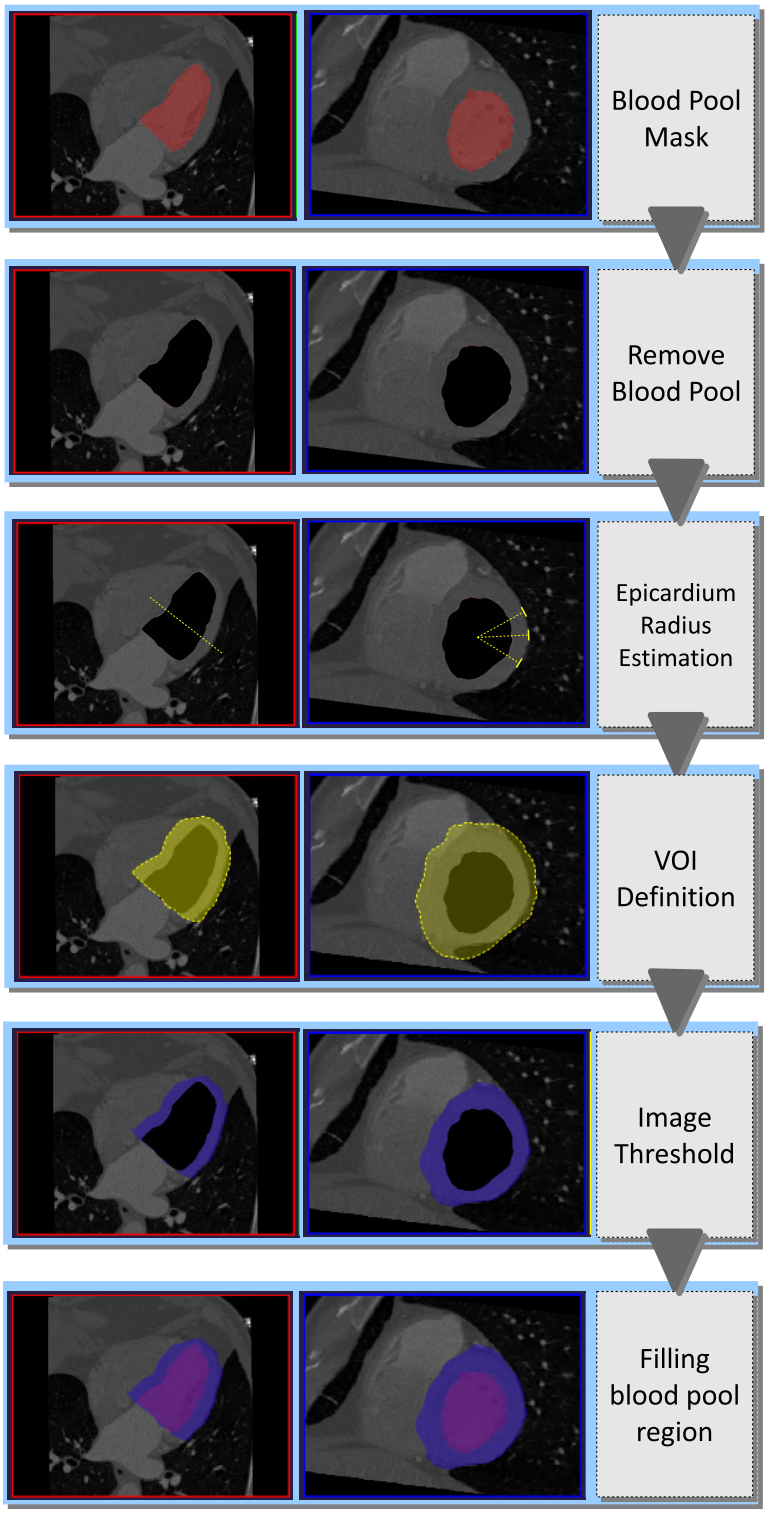


Figure 5.6 – Main steps for epicardium segmentation from the reference cardiac phase.

of epicardium radius and 2) scaling the endocardium mask accordingly and applying it to the image volume.

In the lateral region, where the myocardium is close to the lung, it is easier to detect the epicardium due to the clear transition between myocardium and air. Therefore, to estimate epicardium radius, a short-axis slice at a mid-ventricular level is used and the radius measured. Then, the segmented blood pool is scaled in order to encompass the region expected to contain the epicardium. Nevertheless, this method considers LV symmetry, which is not completely true. For example, the septal wall of the myocardium might be thicker and therefore the epicardium will have a higher radius there. So, the used radius is empirically set as slightly higher than the estimated value.

After defining the region of interest, image threshold is applied. Since there is some inter-patient variation on the attenuation range of the myocardium a simple adaptation step is performed by sampling points inside the region of interest in order to define a mean attenuation value.

Users can customize both the region of interest and the attenuation range used to perform image threshold by using sliders. This is performed while visualizing a low-resolution version of the resulting segmentation.

Finally, a high-resolution version of the segmentation is obtained. Since the goal is, as explained earlier, to segment the region inside the epicardium the blood pool segmentation is added and hole filling is performed.

Figure 5.6 depicts the main steps for epicardium segmentation for the reference cardiac phase.

5.5 Full Exam Segmentation

As explained earlier, poor image quality can be an issue when dealing with the different cardiac phases available. The segmentations obtained for the reference phase are used as a starting point to perform the segmentation of the remaining phases. After obtaining the desired cardiac phase from the DICOM files, image reformat is performed in order to allow analysis along the LV long axis.

5.5.1 Endocardium

The endocardium segmentation for the reference phase is used to define a volume-of-interest (VOI). This completely identifies the region where the blood pool is (and where it should end, at mitral valve level) and there is no need to perform an endocardium location step. It follows image threshold and a hole filling step to include the papillary muscles inside the blood pool.

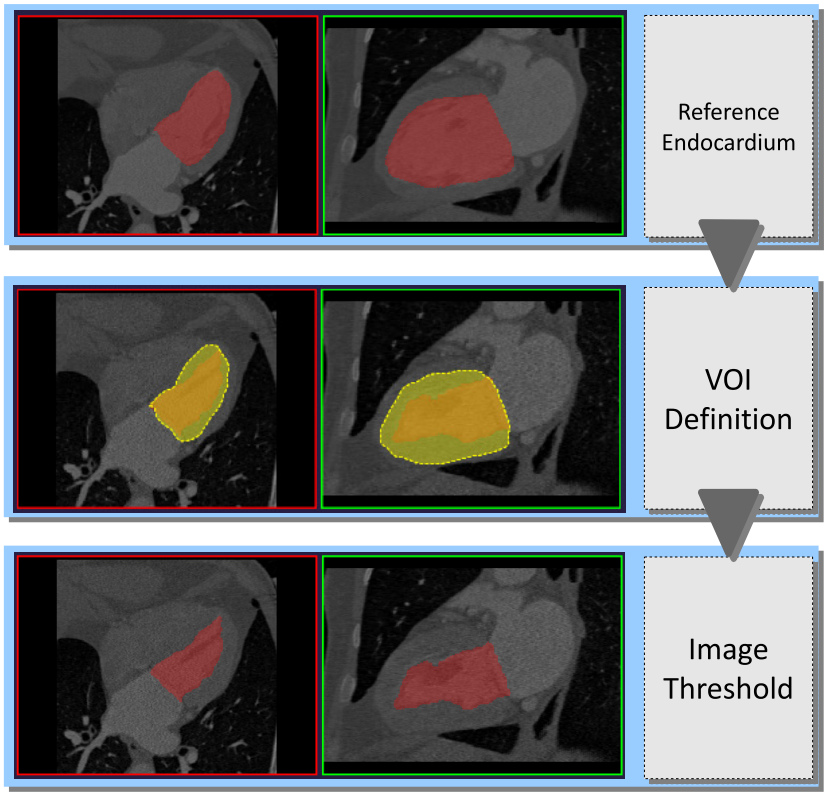


Figure 5.7 – Main steps for endocardium segmentation, having a segmentation for the reference cardiac phase.

Figure 5.7 depicts the main steps for endocardium segmentation for all cardiac phases.

5.5.2 Epicardium

As with the reference phase, the first step is to remove the corresponding blood pool, segmented earlier, in order to prevent the papillary muscles from being considered part of the myocardium.

The epicardium does not vary much along the cardiac cycle. Therefore, the binary mask resulting from epicardium segmentation for the reference phase is used to define a VOI, after slight scaling to cope with LV larger size at end-diastole.

Then, since we wish to segment the region contained inside the epicardium, image threshold is applied and the resulting binary mask is added to the blood pool mask. Finally, hole filling is applied to ensure there are no holes in the resulting binary mask.

5.6 Prototyping

In an earlier development stage, in order to get acquainted with the images and test different ideas for the pre-processing and segmentation method, a prototype was built using MeVisLab[2]. As explained in appendix A, different features turn MeVisLab into a good prototyping platform: off-the-shelf modules for 2D and 3D visualization, integrated support for a large number of ITK filters and the possibility of building custom modules.

The following sections provide an overview of this prototyping stage.

5.6.1 Data Access, Pre-processing and Visualization

MeVisLab processing modules do not work directly with the DICOM format. Instead, an image format is used which optimizes (speeds) the way data can be processed by providing a true 3D image file format. Nevertheless, this is no major difficulty because a module is provided, called DICOMIMPORT, which allows DICOM files to be imported into MeVisLab format and supporting different features such as multiple timepoints import, which was important in our case.

Visualization of the volume sets was accomplished by using the ORTHOVIEW2D (see figure 5.8) module which provides the usual synchronized orthogonal views (axial, sagittal and coronal) and allows cycling through the different slices (and cardiac phases, if needed).

Given the large size of each exam (because it includes 10 or more cardiac phases), in an early stage only one cardiac phase was processed. This can be very useful since, when experimenting methods for noise removal, for example, it would take quite a long time to apply them to all the cardiac phases. The module SUBIMAGE (figure 5.8) was used to extract the cardiac phase of interest and can even be used to extract a particular volume of interest (VOI).

5.6.2 Segmentation

Even though MeVisLab provides several modules which support some kind of segmentation (e.g., thresholding and region growing), given the complexity of semi-automatically segmenting the left ventricle (e.g., localization and orientation) a custom module was developed. MeVisLab's module wizard was used to create the module template and a new module developed using ITK. Apart from a simple initial image format conversion, between the internal MeVisLab format and ITK (and later conversion back to MeVisLab format), programming the algorithm using ITK was completely independent of using MeVisLab. In fact, some previously developed ITK code could be easily integrated.

[2]MevisLab: `http://www.mevislab.de`

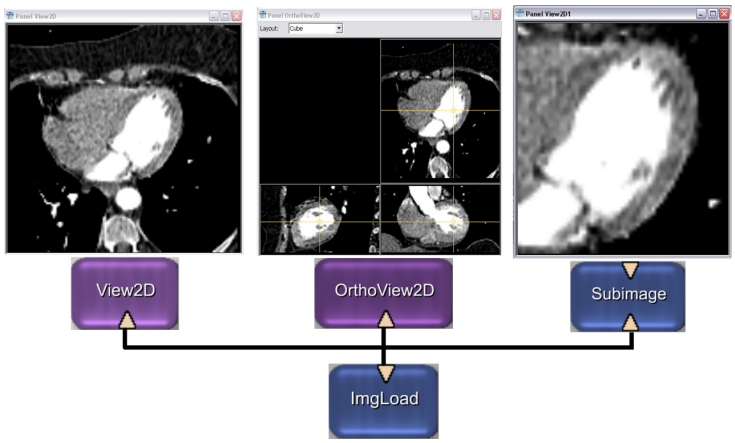


Figure 5.8 – MeVisLab modules providing off-the-shelf image data visualization: slice-by-slice viewing, synchronized orthogonal views and sub-imaging module returning just the defined volume of interest.

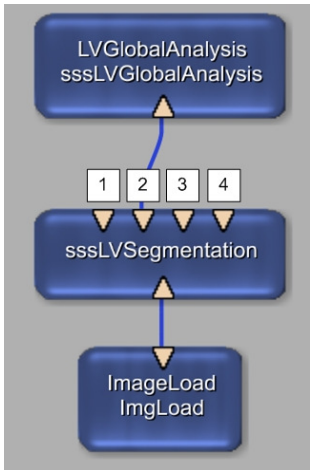


Figure 5.9 – Simple network containing the developed MeVisLab modules (sssLVSegmentation and sssLVGlobalAnalysis). The segmentation module has four outputs: 1) Segmented endocardium for reference phase; 2) Segmented endocardium for all exam phases; 3) Segmented pericardium for all exam phases; 4) Reference phase with left ventricle oriented normally to the coronal plane. The top module concerns global analysis of parameters such as the ejection fraction.

Figure 5.9 shows the developed modules. The SSSLVSEGMENTATION module is responsible for left ventricle segmentation. Apart from the two outputs concerning the segmented regions (endocardium and epicardium) another two have been added. They allow quick access to intermediate segmentation results for debug purposes: one provides the endocardium segmentation for the reference phase and the other provides the reference phase after being oriented according to the results obtained from the long axis estimation step. If everything went well, the left

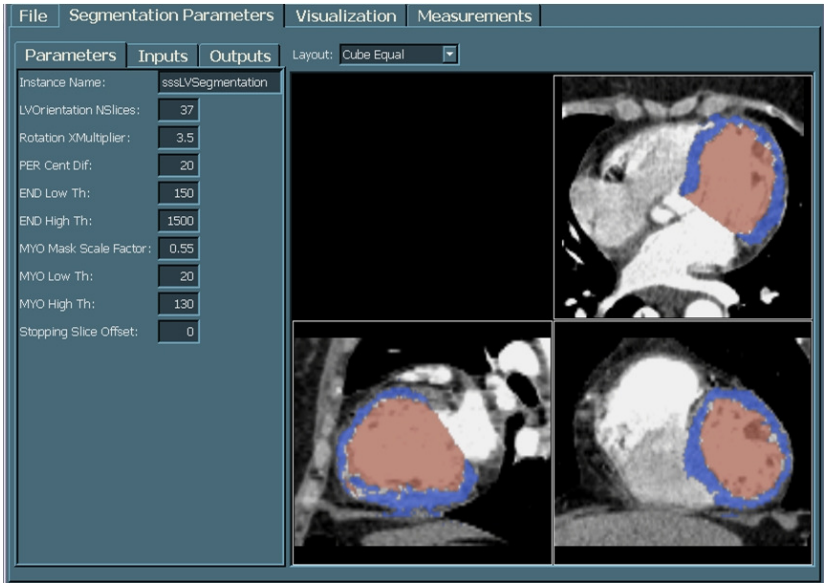


Figure 5.10 – User interface detail showing a tab where the reference phase segmentation results can be viewed and several parameters adjusted.

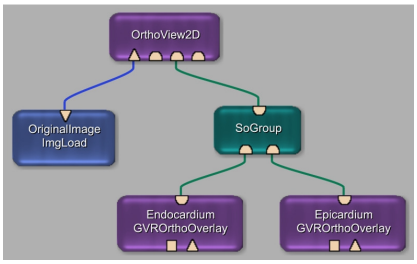
ventricle should appear with its long axis oriented normally to the coronal plane. The module named SSSLVGLOBALANALYSIS takes the segmented endocardium and computes blood volumes, automatically detects the end-systolic (lowest blood volume) and end-diastolic (highest blood volume) phases and computes the ejection fraction [62], a global parameter often used to assess left ventricle function.

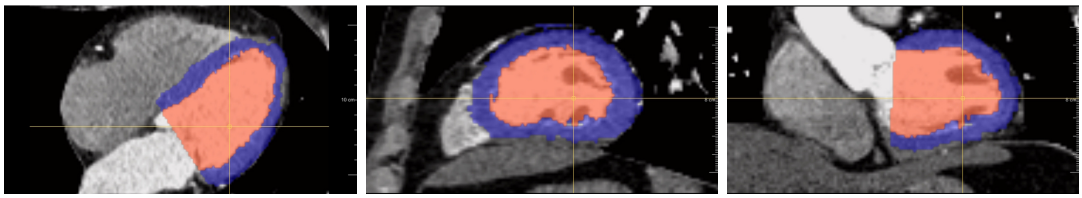
A user interface was built, using MeVisLab functionality, which allows changing some of the main parameters (e.g., threshold levels) while viewing the segmentation results for the 60% phase (figure 5.10), view (with optional animation) the segmentation results for the remaining phases and have access to the computed ejection fraction.

5.6.3 Results Visualization

During development, intermediate results had to be visualized for debug purposes and it was important to provide a context to the obtained segmentations by presenting them over the original images. For that purpose, module ORTHOVIEW2D was used together with overlay features provided by the GVRORTHOOVERLAY module. Figure 5.11 shows the basic network to visualize the image data set using three orthogonal views with the segmented regions overlaid on the image.

Using module VIEW3D a simple three-dimensional view of the segmented data was also built. Figure 5.12 presents several 3D visualizations of the same left ventricle in different cardiac phases.





Axial Sagittal Coronal

Figure 5.11 – Top, basic module network to accomplish image visualization with the segmented regions shown over the CT image using transparency; bottom, three orthogonal views of a cardiac CT volume and corresponding segmentations.

5.7 Qualitative Evaluation

On an early development stage, after developing the first prototype, we opted for a preliminary qualitative evaluation that would allow the detection of any serious segmentation problem and inform further fine tuning of the proposed algorithm.

Such a preliminary evaluation has been performed with the help of three radiographers with 3 to 4 years daily experience in analysing MDCT heart images.

5.7.1 Methods

Even though our method segments all exam phases, in this evaluation only three phases were evaluated per exam: the end-systole and end-diastole, which are used to compute important parameters such as the ejection fraction [51], and the 60% phase which was used as a reference for the segmentation of all remaining phases.

An important aspect of every segmentation method is that the resulting first segmentation (i.e., without manual parameter tweaking or edition) should be as accurate as possible in order to minimize correction time. Having that in mind we decided to evaluate the segmentation results using only the default values for the different parameters (e.g., ratio between reference endocardium radius and current slice radius to determine mitral valve plane; determined empirically during development) and no manual edition.

Seven exams were randomly selected from that week's exams and the endocardium and epicardium were segmented for the 60%, end-systolic and end-diastolic phases. Care was only taken

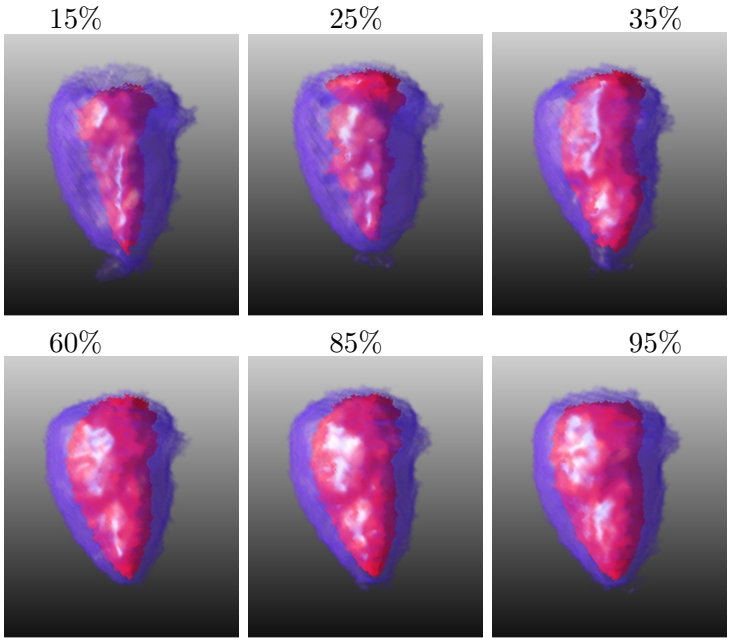


Figure 5.12 – Segmentations viewed using 3D models, obtained using *surface rendering*, and presenting the endocardium in red and the epicardium in blue with a degree of transparency.

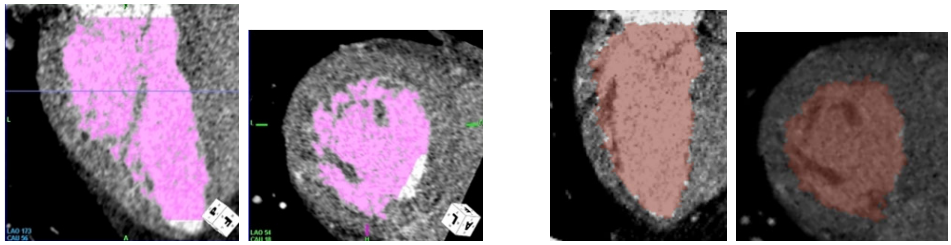
to ensure that no serious movement artefacts were present in any of the exams which might influence the results.

These first segmentations were shown to the radiographers and were considered good first approaches with no major abnormalities and comparable to first segmentations (specially for the endocardium) provided, for example, by a TeraRecon Aquarius[3] workstation which additionally requires a priori manual alignment of the data in three different planes. Figure 5.13 shows an interesting example where, even with the three plane initialization, the TeraRecon workstation provides a worst first segmentation. Notice the segmentation problems in the apex and midventricular slices.

After such preliminary results we decided to perform a more thorough evaluation by asking the radiographers to classify the resulting segmentations as if they were final (i.e., usable for diagnosis purposes). To support the evaluation process it was necessary to allow the following operations:

- Set proper viewing planes according to the planes usually used by the radiographers to perform image analysis of the left ventricle, which are different from the usual orthogonal viewing planes.

[3]TeraRecon Aquarius: `http://www.terarecon.com/products/aq_ws_prod.html`



(a) (b)

Figure 5.13 – (a) Unedited segmentation results provided by a TeraRecon Aquarius workstation; and (b) unedited segmentation results, for the same exam, provided by our method.

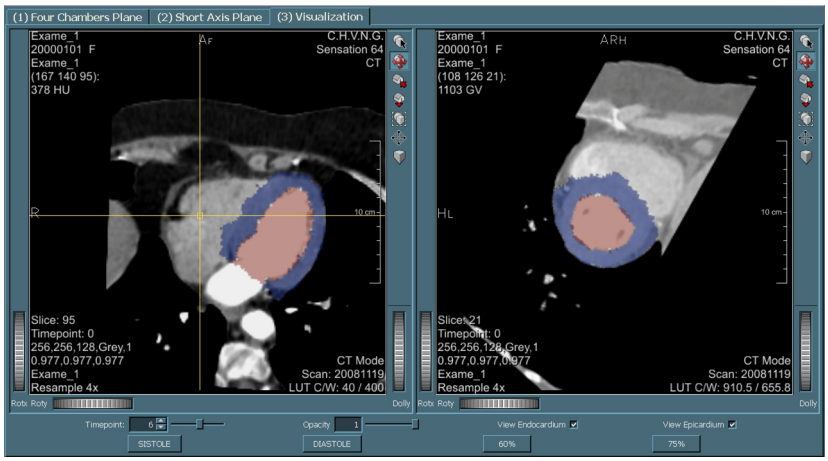


Figure 5.14 – User interface of the application developed using MeVisLab to support the preliminary evaluation process. Endocardium and LV wall segmentations are visualized in a 4-chambers view (on the left) and in a LV short-axis view (on the right) defined by the radiographer on the first evaluation step.

- Visualize the image volume sets corresponding to the end-systole (maximum contraction of the left ventricle) and end-diastole (no contraction) and easily alternate between the two for comparative assessment of the segmentation in important regions such as the outflow tract.
- Scroll along the normal vector of each cutting plane to properly assess the segmentation in different slices.
- Change the opacity of the segmentation overlays to properly analyse the underlying regions.

A simple user interface (see Figure 5.14) was developed using MeVisLab in order to provide the features listed above and guide the radiographers along the evaluation process.

Each radiographer received a grid to guide the evaluation process and where the evaluations should be registered for each of the 28 segmentations (7 exams $\times$ 2 cardiac phases $\times$ 2

(endocardium and epicardium)). Regarding the endocardium, the grid considered segmentation evaluation for four anatomical regions deemed of interest: apex, mid-ventricular slices, mitral valve and outflow tract. For each region the segmentation could be generically classified as OK (optimum segmentation), EXCESS (the segmented region is larger than the ideal) and SHORTAGE (the segmented region is smaller than the ideal). For the last two cases, the radiographer had to classify the severity of the problem using a three-level scale (1 – low significance: the segmentation is very good, although, for the sake of perfectness, it could include/exclude a very small region; 2 – moderate: the segmentation is good but it could be significantly improved by the inclusion/exclusion of a small region; 3 – serious: the segmentation cannot be used without the inclusion/exclusion of important regions). For the epicardium a similar evaluation scale was used and five anatomical regions were considered: apex and the lateral (external) and septal (between ventricles) regions for the midventricular slices and base.

Before evaluating the segmentations for each exam, the radiographers were allowed to adjust view planes in order to define what they considered the proper 4-chamber (ventricles + atria) and LV short-axis views. Next, they were presented with those two views of the different exam phases, with the segmentation superimposed onto them, and allowed to scroll along the slices parallel to the cutting planes (which could be changed, at any time, to encompass their needs). The opacity of the segmented regions could be changed in order, e.g., to better examine the regions bellow the segmentation mask. The endocardium and epicardium segmentations could be visualized separately or at the same time. The radiographers could take all the time they needed to evaluate each phase and were allowed to alternate between phases for comparison (e.g., in order to assess if the papillary muscles were coherently included/excluded).

5.7.2 Results

Table 5.1A shows evaluation results concerning endocardium segmentation. These results are the lowest classification given by the radiographers for each region. Notice that for most exams and cardiac phases the apex and midventricular slices where considered to be well segmented. Concerning the mitral valve and outflow tract there are different problems with different severities. A worst case example is exam 2 with level 3 problems in four situations. Nevertheless, notice that this just means a segmentation that must be edited before usage and not a completely abnormal segmentation. Figure 5.15a shows the endocardium segmentation for the end-systole of exam 2 as an example of the worst segmentation found for the mitral valve region. Figure 5.15b shows the corresponding automatic segmentation as provided by a Siemens Circulation[4] workstation with a significant region wrongly segmented inside the left atrium.

[4]Siemens syngo Circulation: `https://www.medical.siemens.com/`.

Table 5.1 – Evaluation results for endocardium and epicardium segmentation.

A. Endocardium Segmentation Evaluation

	End-Sistole				End-Diastole				60%			
Exam	Apex	Mid.	Valve	Tract	Apex	Mid.	Valve	Tract	Apex	Mid.	Valve	Tract
1	O	O	-	O	O	O	O	O	O	O	+	O
2	O	O	+++	+++	O	O	++	++	O	O	+++	+++
3	O	O	+	+++	O	O	O	++	O	O	++	++
4	O	O	O	+	O	O	- - -	O	O	O	- -	O
5	-	-	+++	+++	-	O	- -	- - -	-	O	+++	- - -
6	O	O	++	- - -	-	O	++	- - -	-	O	+++	- - -
7	-	- - -	- - -	- -	-	- - -	- -	- - -	- -	- -	++	- - -

B. Epicardium Segmentation Evaluation

	End-Systole					End-Diastole				
Exam	Apex	Mid.		Base		Apex	Mid.		Base	
		L	S	L	S		L	S	L	S
1	O	O	+++	O	+++	O	O	+++	O	+++
2	- -	O	O	O	+++	- -	O	O	O	O
3	-	O	O	O	+	-	-	-	-	-
4	O	-	+++	+++	O	O	-	++	O	O
5	- -	- -	++	++	++	- - -	+++	+++	- - -	- - -
6	- -	- -	+++	O	O	- -	-	+++	- -	O
7	- - -	- - -	++	- -	- -	- - -	- - -	-	-	-

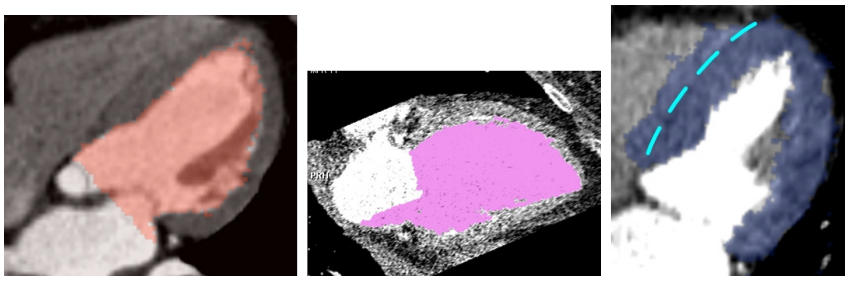
O : Optimum +: Excess – : Shortage

Number of symbols conveys severity level, e.g.: +++ : level 3 excess; – – : level 2 shortage

Remarkably, during the evaluation, we noticed that two different criteria might be used to evaluate outflow tract segmentation: one considering that the tract should not be included in left ventricle segmentation and another considering that all the tract up to the aortic valve should be considered. These different (acceptable) criteria might be a source of variability between radiographers as confirmed later during the evaluation presented in chapter 8.

Concerning epicardium segmentation (lowest evaluation results, among the three radiographers, presented in Table 5.1B), the radiographers found most of the severe problems in the septal sections of both midventricular and basal slices. Even though our method tries to adapt the threshold interval to each phase and seems to work very reasonably for some exams, the attenuation values found in the septal section are, in many exam phases (due to noise and low radiation dosage) similar to those found inside the right ventricle. The initial mask applied to the image, to define a restricted region of interest, is limiting the severity of some of the problems but these results showed that epicardium segmentation clearly needed further improvements. At this time, the segmentation method did not include epicardium radius estimation which was adopted after these results to reduce the problem.

Figure 5.15c shows an example of a segmentation problem found in the septal section. The segmentation should roughly end in the dashed line.



(a) (b) (c)

Figure 5.15 – Endocardium segmentation, as provided by our method (a), for the end-diastolic phase of exam 2 and segmentation as provided by a Siemens Circulation workstation (b). In (c), example of epicardium segmentation problem found in the septal section.

5.8 CardioAnalyzer

The prototype built using MeVisLab provided a good starting point for experimenting with different ideas for LV segmentation. Nevertheless, taking into account the main goals of our work, support is needed for developing different interaction patterns, include and manage multiple data from different analysis methods and control system performance by being able to work at different application levels from image opening to visualization. Therefore, even though such features might be developed, to some extent, using MeVisLab and module customization, this would not allow full control over the developed tool and would require a considerable effort. Such effort is also present in developing a new framework but results in better performance and versatility.

Following on the main conclusions presented in appendix A we chose the Medical Interaction Toolkit (MITK) to provide the core features (data handling and image visualization) of our framework, Qt for user-interface development and ITK for image processing.

A software application was developed, CardioAnalyzer, in order to provide a framework to support the work carried out.

Figure 5.16 presents a general view of CardioAnalyzer's user interface showing the image visualization windows and, to the right, an area where the available features regarding the current segmentation task are presented.

An overview of CardioAnalyzer, including the main features supporting LV segmentation (e.g., the different segmentation stages), can be found in appendix B.

5.9 Conclusions

This chapter presents a left-ventricle segmentation method to be applied to cardiac CTA images. It starts by segmentation of the LV for a reference cardiac phase and then uses it to define volumes

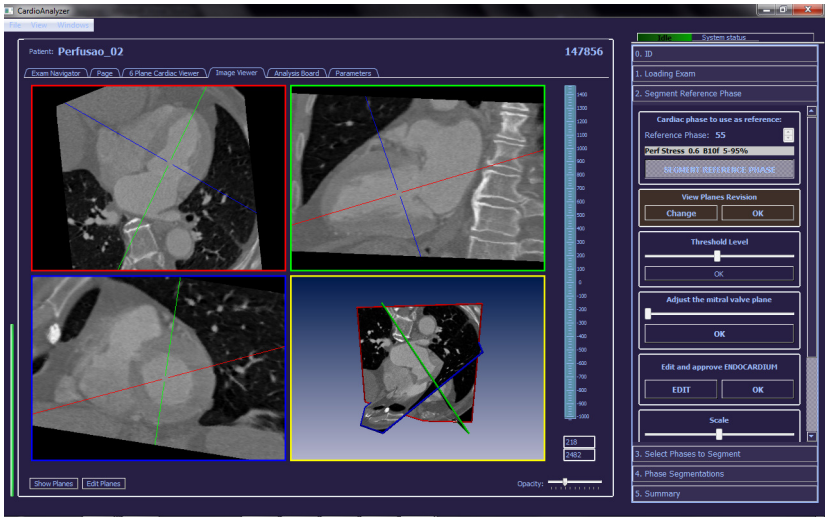


Figure 5.16 – General view of CardioAnalyzer showing different views of a CTA exam and interface widgets for LV segmentation support.

of interest (VOI) to support the segmentation of the remaining cardiac phases.

From the data collected during the preliminary evaluation study presented in section 5.7 we concluded that the detected issues might be better solved by providing an editing tool (presented in chapter 6) than trying to improve the segmentation method.

At this stage, the presented segmentation method already fullfills three of the four requisites set before development (see Introduction, pg. 51). The fourth requisite, concerning validation of the method by radiographers, will be the subject matter of chapter 8.

Chapter 6

Editing Tool for 3D Segmentations

As important as segmentation methods are the tools provided to allow users to guide them and/or correct their outputs. The MITK library already provides a segmentation editing tool. Unfortunately, it only supports 2D editing. Given that the left ventricle spans over several slices, any editing would imply modifications to a large number of them and therefore result in a time-consuming task.

Consequently, in addition to the segmentation method presented in chapter 5, a 3D editing tool is proposed, developed considering the shape of the segmentations and expected editing tasks.

This chapter describes different approaches tested for 3D segmentation editing, the main advantages/disadvantages found in each and a simple assessment concerning how a 3D editing tool allowed reducing the time required to perform two editing tasks, when compared to a 2D tool.

Publications:

The work presented in this chapter has been partially published in:

- Samuel Silva, Beatriz Sousa Santos, Joaquim Madeira, Augusto Silva, "A 3D Tool for Left Ventricle Segmentation Editing", Proc. Int. Conference on Image Analysis and Recognition (ICIAR 2010), LNCS 6112, pp. 79–88, Póvoa do Varzim, Portugal, 2010

6.1 Introduction

Image segmentation is a common step in numerous application areas. Segmentation in medical imaging is quite challenging as completely automatic methods fail to work in every possible situation due to the natural biological variation. Therefore, as important as image segmentation algorithms are the tools which allow user interaction (a survey can be found in Olabarriaga et al. [5]) to guide the method or correct the results. Segmentation editing is one of those features which quite often do not deserve much attention as the focus is, in general, on the automatic steps of the segmentation algorithm [63]. Even though, in practice, editing tools are provided for all segmentation tasks, their suitability is sometimes neglected and, although they allow corrections to be performed, they require a large amount of work and time. One such example is editing 3D segmented regions on a slice-by-slice basis by contour manipulation [30].

Following on the work presented in chapter 5, concerning LV segmentation from CTA exams, the provided segmentations sometimes need to be corrected. In the presence of poor segmentations, some high level parameters can be used to correct them but the most common problems can be easily solved with some editing (as reported by radiographers on a preliminary evaluation described earlier). Thus, an editing tool must be provided which is easy and intuitive to use and is well suited for the task at hand. This must consider the 3D characteristics of the segmented LV and the kind of LV analysis performed by the radiographers using view planes which differ from the usual orthogonal planes (axial, sagittal and coronal).

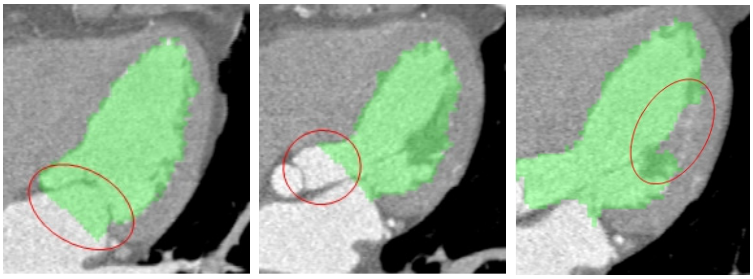
The MITK library already provides a segmentation editing tool which presents two limitations: it only supports editing on the usual orthogonal planes[1] (axial, sagittal and coronal) and is a 2D tool (i.e., editing only possible slice-by-slice). Since each image volume used for LV segmentation is approximately $512 \times 512 \times 256$ and the LV spans over a considerable number of slices (around 100 short-axis slices), it would be very tiresome to perform slice-by-slice segmentation editing.

Editing is also important in our case since, for each cardiac exam, the left ventricle must be segmented for 12 phases along the cardiac cycle (i.e., 12 image volumes taken from systole to diastole): in case of a segmentation problem, if the user is allowed to edit a first automatic segmentation of one of those phases, that information can be used to improve the remaining 11 segmentations, therefore reducing the amount of user intervention needed.

We present a 3D tool for LV segmentation editing of first segmentations provided by the developed segmentation method. This tool is included in the segmentation protocol used to obtain validated LV data for analysis.

This chapter describes different approaches tested for 3D segmentation editing, the main advantages/disadvantages found in each and a simple assessment concerning how a 3D editing

[1] At the time of writing this thesis MITK already supported 2D editing on oblique planes.



(a) (b) (c)

Figure 6.1 – Common segmentation problems detected in the automatic segmentation method output: a) segmentation beyond the mitral valve; b) outgoing tract not included in the segmentation and c) papillary muscle not included in the blood pool.

tool allowed reducing the time required to perform two editing tasks when compared to a 2D tool.

6.2 3D Editing Tool

When designing the presented 3D editing tool two main goals were taken into consideration: first, the tool needed to allow easy correction of the typically detected problems (see figure 6.1 for some examples), and second, the tool should support real-time interaction. This second goal limited the complexity of the operations performed during editing in order to minimize the associated computational cost. Furthermore, the tool should provide editing in any view plane and work by using the mouse as interaction device.

Two different editing tools were developed: one based on voxel mask editing and the other based on surface editing.

6.2.1 Voxel Mask Editing

This first method works at voxel level by adding/removing voxels from the segmented region. The editing brush has a spherical shape thus providing 3D editing.

Two editing modes are available: ADD voxels to region and REMOVE voxels from region. Editing mode selection is performed automatically according to the voxel value at the centre of the brush when the editing operation is started. If it is an active voxel (i.e., a voxel part of the current segmented region), the tool is set to ADD mode (figure 6.3a) and if it is an inactive voxel (i.e., not part of the current segmented region) the tool is set to REMOVE mode (figure 6.3b). The tool keeps the current mode until the editing step is finished (i.e., mouse button is released). This automatic selection mode is based on the observation that when the user wants to add voxels to the border of the segmented volume it is natural to start the editing from inside the object and the contrary, i.e., from outside the object, when the goal is to remove voxels. This also guarantees

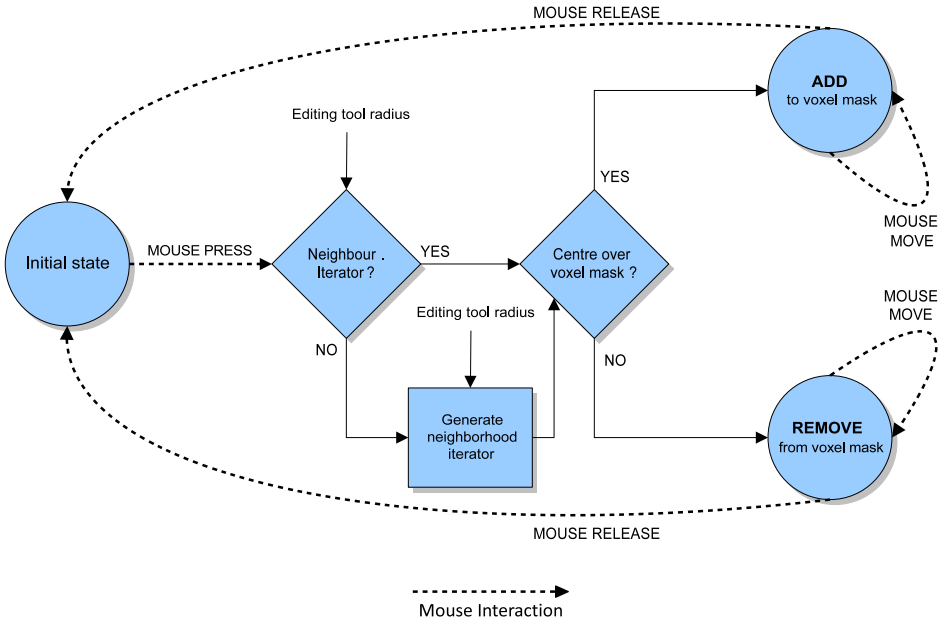


Figure 6.2 – Diagram depicting voxel editing steps and automatic editing mode (ADD or REMOVE) choice.

(although some awkward situations may arise) some continuity of the segmented region. Since we know the left ventricle is a closed region the user must always travel from within it to where he wants to add voxels. This does not, however, solve the problem of isolated regions resulting from voxel removal around them which is dealt, in our case, by a post-processing step.

Figure 6.2 depicts the main steps of the voxel editing operation.

To speed the voxel mask modification operation a neighbourhood iterator is used, provided by the ITK library. It allows the definition of a 3D neighbourhood region (a cube centred on the desired voxel) in which it is possible to activate only the neighbourhood voxels we are interested in visiting. In our case, voxels are activated to obtain a sphere shaped neighbourhood (according to the editing tool radius chosen). Since the neighbourhood is defined using offsets towards the central voxel it can be easily relocated without having to re-set the active neighbourhood voxels. Therefore, the iterator initialization needs only to be performed once for each tool radius desired.

The tool radius can be chosen (using two keyboard shortcuts, to increase and decrease the radius) from a limited set of values established based on the characteristics of the left ventricle and image resolution. Given that there are no particularly thin regions there is no need for very small radius tools. It should only be guaranteed that a small enough tool exists to be used on the outgoing tract. On the other hand, a large radius tool could be used to edit the segmentation on the mitral valve region but, if image resolution is high, the number of voxels in the neighbourhood starts to increase rapidly with the radius, thus influencing both the initialization time of the tool

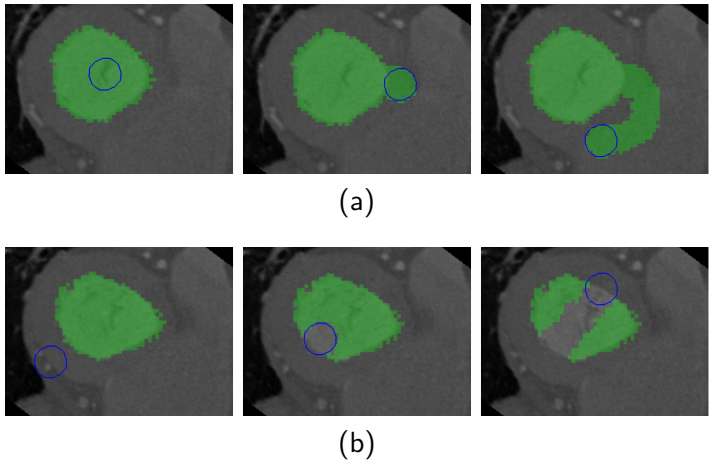


Figure 6.3 – Two modes available for bit mask editing. In the first mode (a), ADD, the editing operation starts on an active region and voxels are added to the object as the tool is moved. In the second mode (b), REMOVE, the editing operation starts on an inactive region and voxels are removed as the tool is moved.

(to set the spherical neighbourhood) and its interactivity during editing.

6.2.2 Surface Editing

Another option for editing is to work with the surface of the segmented region obtained, for example, by using the marching cubes algorithm [64] provided by the VTK library [65] thus obtaining a polygonal representation of the surface. This has the advantage that it does not occlude the image with the segmentation mask as much as the voxel mask and does not suffer from isolated regions being created when separated from the main region by removing voxels in-between. Intersecting the surface of the segmented region with different view planes produces contours which can be intuitively adjusted by the user.

Notice that generating contours (instead of a surface) would also be possible by applying an image cutter to the voxel volume with the desired orientation and extracting the contours but, as the user must be able to choose and change to any view plane she/he finds necessary, it would result in a method of much higher complexity.

In this case, the editing tool consists of a sphere, moved by the user, with its centre at cursor position, which deforms the polygonal surface when pressed against it. It appears in each visualization plane as a circle (see figure 6.4) and can be used freely in any of them. Surface deformation is obtained by changing the polygonal surface vertices position in order to keep them always at no less than sphere radius distance from the tool centre (i.e., outside the sphere).

When the editing sphere approaches the surface, the closest vertex (if any at less than the sphere radius distance) to the sphere centre is determined and all its neighbours which are also

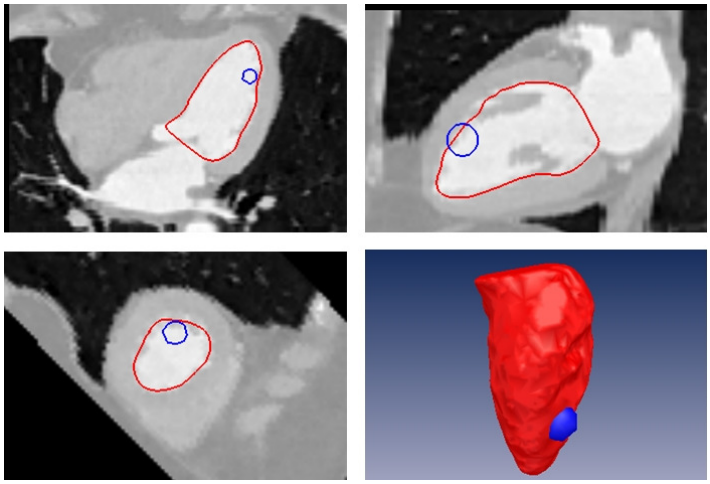


Figure 6.4 – Segmentation editing by surface deformation. The editing tool is a sphere and can be used in any of the view planes. It appears in different sizes depending where the view plane intersects the sphere. On the bottom right image a 3D view of the surface (in red) and sphere (in blue) are shown

found inside the sphere are determined. Since computing vertex neighbourhoods has a high computational cost, particularly if the sphere radius is large, the first seven neighbourhoods (one-ring, two-ring, ...) are pre-computed and stored in memory. This has also the advantage of being a lot faster than the individual neighbourhood computations since previously computed neighbourhoods for some of the vertices can be used to speed the computation of the remaining neighbourhoods.

After finding the list of vertices that must be displaced it is necessary to determine the direction of their displacement. A simple idea is to displace them radially, i.e., using the normals to the sphere surface (figure 6.5a). This method has the disadvantage of creating a low density of vertices on the surface region which suffers the largest displacement. It is easy to understand that, very rapidly, the vertex density in that region will be so low that further displacements will be impossible. It is also important to notice that with only a few vertices it is harder to properly deform the surface to accurately contain the desired region. This method will also easily allow surface self intersections as the vertices will move freely in every possible direction.

Thus, it is important to design a vertex displacement method which reasonably preserves vertex density and reduces the chance of surface self-intersections, at least for small correcting operations. This was accomplished by displacing the vertices along the corresponding surface normal, i.e., the vertex is always moved following the direction of the original surface normal at its position. As can be seen in figure 6.5, this method results in better vertex density preservation and is less prone to surface self-intersection although sharp edges might be a problem. As the

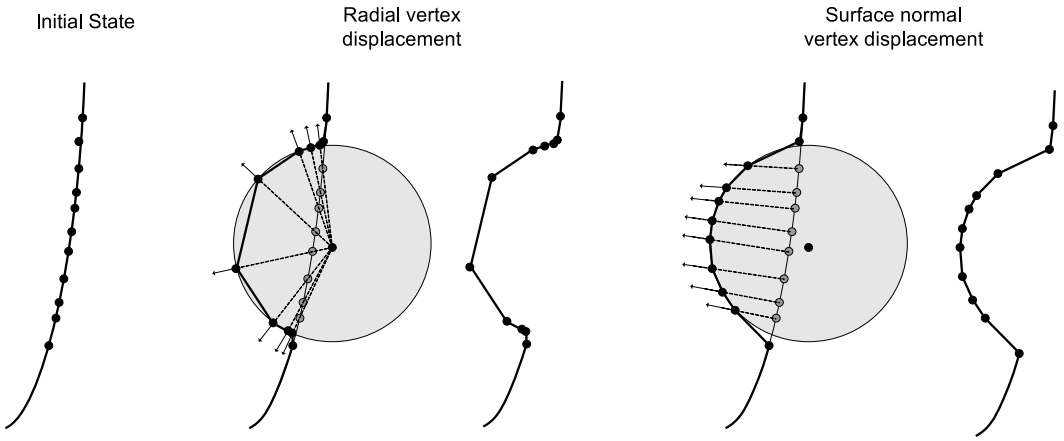


Figure 6.5 – Two different options for defining vertex displacement direction: on the left, vertices are displaced radially in a direction defined by the sphere centre and their initial position resulting in vertex dispersion; on the right, vertices are displaced in the direction given by the original surface normal at their original position which keeps vertex density.

vertices always move in the same direction it is possible to easily revert any of the editing operations by applying the sphere from the opposite side.

Since the vertex neighbourhoods are pre-computed the impact of a large sphere radius on interactivity is small (within reasonable radius limits) and, thus, the user is allowed to use a wide range of radius values suited for each situation as illustrated in figure 6.6.

It is important to notice that in our particular situation the left ventricle will never need significant corrections: if a really bad segmentation is performed the user can change some high level parameters during segmentation (e.g., mitral valve plane) to obtain better results before editing. For directly editing really bad segmentations this method would be harder to use, e.g., if the surface had to be corrected to a much larger or less smooth surface, as the number of vertices is fixed.

6.2.3 Mixed Editing

The methods presented above can also be used interchangeably, if necessary, although changing between them implies some delay (around 4 seconds on a computer with a 2.2GHz dual-core CPU). If initially editing the voxel mask, when changing to surface editing, a new surface must be generated from the current voxel mask and vertex neighbourhoods must be computed. When reverting from surface editing to voxel editing the voxel mask contained inside the current surface must be computed.

Computing a new surface from the current voxel volume is accomplished, as previously men-

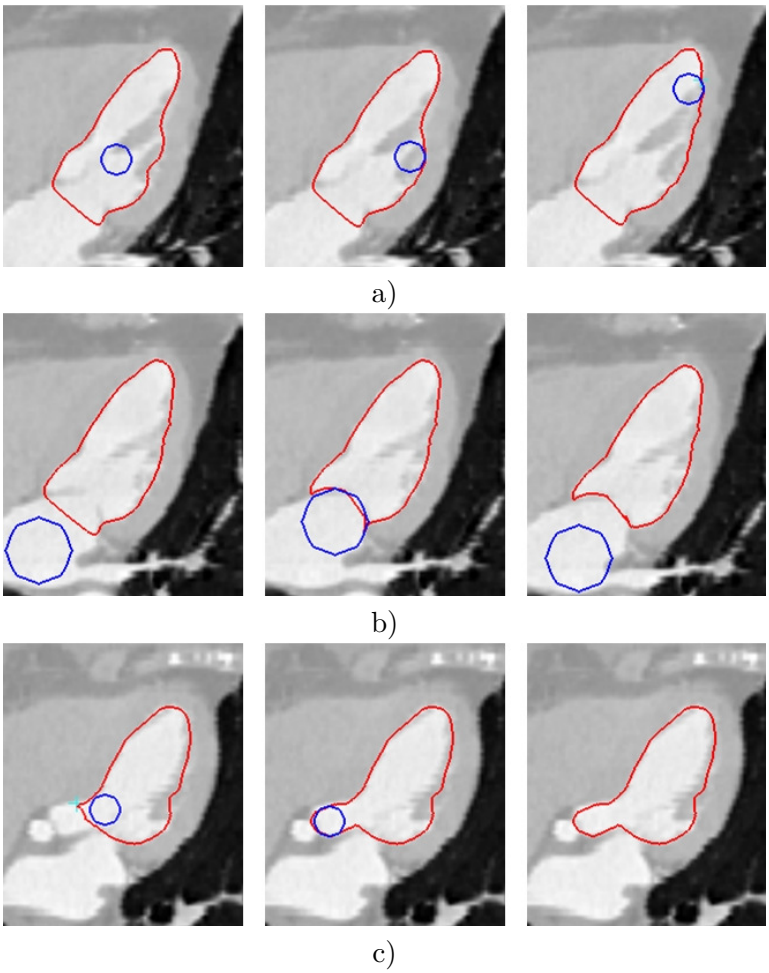
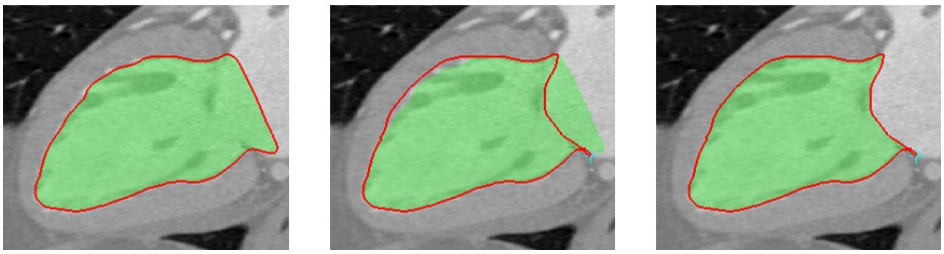


Figure 6.6 – Different editing situations using surface deformation: a) Correction of the segmentation to include the papillary muscles inside the surface; b) a larger tool radius has been chosen to correct the segmentation on the mitral valve region and c) tool radius has been reduced to an adequate size to correct the outgoing tract.

tioned, by using the marching cubes algorithm. To obtain the voxel mask contained inside a surface a method provided by VTK is used to obtain a stencil from the polygonal data which is then applied to a voxel volume the size of the original image, as a "cookie cutter", to obtain the desired result.

Figure 6.7 shows some examples of conversion from voxel mask to polygonal surface and vice-versa.

Computing the voxel volume contained inside a surface is also important to compute features of the segmented region such as regional and global blood volumes (by counting the voxels and multiplying by voxel volume) for left ventricle function analysis.



(a) (b) (c)

Figure 6.7 – Converting from voxel mask to surface and vice-versa: (a) surface generated from the voxel mask using the marching cubes algorithm; (b) the surface is deformed using the presented tool; (c) the voxel mask contained inside the surface is computed.

6.3 Evaluation

To assess the applicability and quantify the advantages of the proposed tool a simple evaluation was conducted concerning the time taken to perform the same editing task using voxel mask (3DV) and surface editing (3DS) and the 2D editing tool provided by MITK.

Considering the common editing operations which will be performed by the radiographers (figure 6.1), two tasks have been chosen for this evaluation: task 1 consisted in adjusting the segmentation mask to the mitral valves which implied removing voxels from the initial segmentation; task 2 consisted in adjusting the segmentation mask to the LV wall adding voxels to the initial segmentation.

Three users participated in this evaluation. In a first stage, they received a short explanation about the different tools and were allowed to use them (user A was allowed to train longer) to get acquainted with the different features. On a second stage, the users were asked to perform both tasks with the three available methods.

As can be observed in table 6.1 the developed tool clearly allows performing the considered tasks in less time than a typical 2D editing tool available in MITK. This shows that with a simple 3D tool, focused on the characteristics of the region and tasks to perform, a considerable gain was attained. For the evaluated tasks the time taken to accomplish them was significantly greater for the 2D tool and smaller for both 3D editing modes, which yield similar task times.

During and after the evaluation users were invited to comment on the different tools. They were unanimous in stating that having to set the editing mode (add or subtract) in the 2D tool was confusing and preferred our tool which sets the mode based on cursor position. Even though times were similar for both 3D tools users preferred voxel editing (3DV). This was probably due to occasional latency in surface editing resulting from a greater computational load.

User A trained longer and managed to accomplish all tasks in less time than the other users.

Table 6.1 – Time (in seconds) taken to complete an editing task using the presented tool (3DV - voxel editing and 3DS - surface editing) and a 2D tool provided by MITK.

User	Task 1			Task 2		
	2D	**3DV**	**3DS**	**2D**	**3DV**	**3DS**
A	236	35	50	333	42	61
B	594	78	99	554	88	121
C	600	105	107	768	119	90

This suggests that, with training, performance with our tool can be further improved to even smaller task times.

6.4 Voxel Editing and Binary Outline

After the evaluation, it was clear that users preferred voxel editing due to its simplicity and better performance when compared with surface editing. Nevertheless, editing a contour instead of a voxel volume can have the advantage of not occluding image contents. Therefore, it would be an advantage if both features could be provided in a single editing method.

After some research concerning non-documented MITK features, this was accomplished by allowing users to change between the voxel mask and its outline representation. This might look just like what is provided by surface editing with the advantage of keeping good interactivity and performance. Then, why should surface editing ever be considered? Notice that surface editing allows much more control over how the deformation is performed, controlling how surface vertices are moved (e.g., direction) or easier post-editing processing (e.g., smoothing) and can, therefore, be a more suitable method in circumstances where a more active control over the kind of "deformation" is required.

6.5 Discussion and Conclusions

This chapter presents a 3D tool for editing left ventricle segmentations. Its purpose is to allow radiographers to rapidly correct first segmentations provided by an automatic segmentation method. This approach, when compared with the usual 2D slice-by-slice method, has the advantage of being faster and more reliable as it reduces the probability of uncorrected slices being left behind and ensures a higher degree of coherence from slice to slice.

The tool has been developed paying attention to the characteristics of the target data (left ventricle segmentations) and necessary editing operations allowing for simplicity and good suitability, thus presenting a smaller semantic distance towards the tasks envisaged by the user.

Figure 6.8 – A worst case example of a problem which might occur after a modification using an editing tool with a very large radius.

Voxel mask editing is very fast, thanks to the neighbourhood operator, and allows good interactivity and results. By applying a post-processing filter at the end it is possible to discard isolated voxels missed by the user.

The surface based method works well for small and medium scale modifications, such as those presented in figure 6.6, but larger modifications or a large number of editing operations around the same region might sometimes result in awkward situations as the one illustrated in figure 6.8. Apart from possible surface self-intersections this can be just a consequence of vertices below the current view plane starting to appear in it, due to a large displacement or their normal having an orientation that favours such an event. Nevertheless, it has a bad impact on users and should be avoided.

A simple evaluation session was conducted and users asked to perform editing tasks with both surface and voxel editing methods and with a 2D method. According to users, voxel editing was clearly the best method because it performed faster and no awkward situation such as surface self intersection occurred. Adding the possibility of showing just the outline of the voxel mask allowed the users to have an option which does not occlude the image (an advantage of surface editing).

Smoothing might improve the results for surface editing and some experiments with simple smooth operations (interactivity must be kept) applied to the affected neighbourhood (or its border) have been performed with some promising results, but it is still not clear when and where to apply them and how to detect problematic regions. So, even though the current surface editing feature is already usable its robustness and performance can still be improved. Voxel mask editing with binary outline representation fully served our goals regarding its technical performance and user satisfaction levels.

One last aspect must also be highlighted. The shape of the editing tool was not chosen blindly. At a first glance, the spherical shape of the editing tool might just seem the "common" approach. Yet, this shape was also chosen because it suits the shape of the region to be edited (LV). Notice that the LV has a smooth outline and, for example, in short-axis slices, it has a circular outline. Furthermore, we must not forget we are editing in 3D. This means that when editing on a particular slice, adjacent slices, above and below, are affected. Using a spherical tool, as opposed, for instance, to a cylindrical tool, ensures that those "invisible slices" are edited to a smaller extent. Nevertheless, editing in 3D using this tool requires some attention to possible

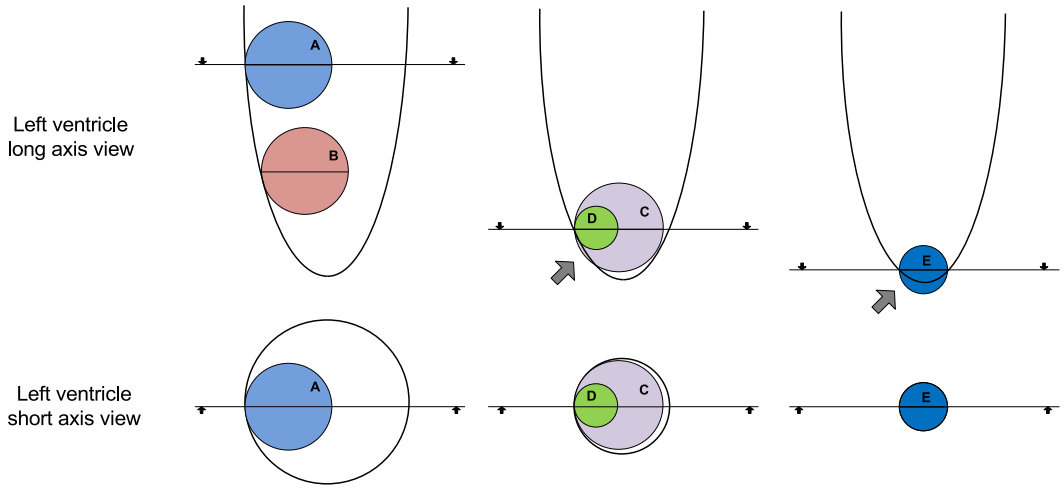


Figure 6.9 – Different editing situations with various editing tool positions and radia. Notice how on the short axis views the tool always seems adequately positioned. On the other hand, if both views are used possible positioning problems can be identified (C and E).

problems. Figure 6.9 depicts several editing situations. Notice that for editing situations A and B, editing tool shape and size are perfectly adequate for editing without going outside the desired LV shape. In situation C, on the other hand, the chosen radius is not adequate but a radius reduction (D) can solve the problem. Finally, in situation E, the tool size and positioning is quite inadequate for editing in that region and will result badly. What this highlights is that, if users are performing editing in 3D their analysis must not restrict to a single view of the data. This would be enough for 2D editing, of a contour, but might result badly in 3D editing.

Even though this tool was developed with a particular application to left ventricle segmentations it can be of use in any situation which deals with similar segmented regions, i.e., with a smooth outline.

Chapter 7

Similarity Measures for Left Ventricle Segmentations Comparison

"All animals are equal but some animals are more equal than others."

— George Orwell, Animal Farm

MEDICAL image processing and analysis tools often include segmentation for quantitative measures of extent, volume and shape. Validation of new segmentation methods and tools usually implies comparing their various outputs among themselves (or with a ground truth), using similarity measures. Several such measures are proposed in the literature but their use is often guided by criteria resulting from their popularity in the literature.

Among the different similarity and discrepancy measures available it is important to select those which are relevant for a particular task, as opposed to using all measures, therefore avoiding additional computational cost and redundancy.

A methodology is proposed which enables the assessment of how different similarity and discrepancy measures behave for a particular task and the selection of those which provide relevant data.

Acknowledgement:

The work presented in this chapter has been performed with the collaboration of Professor Carlos Ferreira from the Department of Economics, Management and Industrial Engineering, University of Aveiro.

Publications:

The work presented in this chapter has been partially published in:

- Samuel Silva, Beatriz Sousa Santos, Carlos Ferreira, Joaquim Madeira, Augusto Silva, "A preparatory study to choose similarity metrics for left-ventricle segmentations comparison", Proc. SPIE Medical Imaging 2011: Computer-Aided Diagnosis, vol. 7963, pp. 796326 – 796334, Orlando, Florida, USA, Feb., 2011

An application of the conclusions presented in this chapter to the comparison of tongue segmentations can be found in:

- Paula Martins, Catarina Oliveira, Samuel Silva, António Teixeira, Augusto Silva, "Tongue Segmentation from MRI images using ITK-SNAP: Preliminary Evaluation", Proc. IADIS Computer Graphics, Visualization, Computer Vision and Image Processing (CGVCVIP) 2011, pp. 3–10, Rome, Italy, 2011

7.1 Introduction

A major concern when performing image segmentation (automatic or manual) is to account for its accuracy and variability: how close is it to the 'real' anatomic region? how much does it differ when performed by different users/methods? and how much does it differ when performed by the same user (or method) at different times?

Evaluation studies are used to assess these questions (further details to be given in chapter 8) by gathering segmentation data and comparing such data among themselves (or with a ground truth), using several similarity measures.

For particular situations some of the measures might not be 'sensible' to the typically expected differences or several of them might just provide equivalent results. Using the largest number of measures might seem the easiest option but it has considerable implications if a large number of segmented volumes must be compared. Therefore, a preparatory study should be performed to assess the behaviour of different measures and decide which ones are relevant to use.

We present a short overview on various similarity measures described in the literature. Then, several of those measures are used to compare volumes obtained using the left ventricle (LV) segmentation tool presented in chapter 5. A statistical analysis involving exploratory data analysis and multivariate techniques is then presented characterizing the behaviour of each measure and highlighting similarities among them.

The main goal of the work presented in this chapter is to assess which of the similarity measures available in the literature might be relevant to use when evaluating our segmentation method (using the evaluation protocol presented in chapter 8).

7.2 Similarity and Discrepancy measures

This section presents a brief overview of measures used for comparison among segmentations. It is focused on area/volume based comparison measures, the kind of data resulting from our segmentation method. Therefore, measures such as the Pratt figure of merit [59], commonly used for contour (edge) comparison, are not mentioned.

Along this section the term similarity is used for measures which increase with decreasing difference between the compared sets, while discrepancy is used for measures which increase with increasing difference between the compared sets.

In what follows, X and Y represent two regions being compared and, when applicable, Y is considered the reference region. Figure 7.1 illustrates the different regions of interest for this discussion. Whenever used, $|Set|$ means the number of elements in Set. For the sake of

X is the region compared with Y (considered the reference).

True positives (TP)	a	$\lvert X \cap Y \rvert$
False positives (FP)	$b1 + b2$	$\lvert X \rvert - \lvert X \cap Y \rvert$
False negatives (FN)	$c1 + c2$	$\lvert Y \rvert - \lvert X \cap Y \rvert$
True negatives (TN)	d	$\overline{\lvert X \cup Y \rvert}$

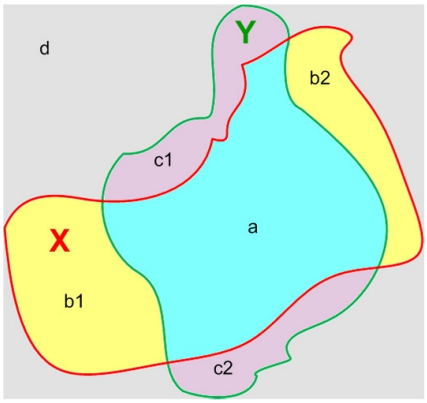


Figure 7.1 – Notable regions of interest when analysing the similarity between two regions X and Y.

simplicity (and as shown in figure 7.1) the various measures will be presented for 2D (pixel sets). Nevertheless, all can be applied to 3D (voxel sets).

One of the simplest similarity measures is the Jaccard [66] coefficient which is equal to one if the compared regions completely overlap and equal to zero if the regions do not overlap and is defined as:

$$Jac = \frac{|X \cap Y|}{|X \cup Y|} \tag{7.1}$$

An equivalent similarity measure, known as Dice coefficient can be expressed as:

$$Dice = \frac{2\,|X \cap Y|}{|X| + |Y|} \tag{7.2}$$

In Babalola et al. [66] two variants of the Dice [66] coefficient are described: the false positive Dice (FP_{Dice}), and the false negative Dice (FN_{Dice}). While the Dice coefficient conveys the level of similarity, these two discrepancy measures allow a more accurate characterization of what has happened and are defined as:

$$FP_{Dice} = \frac{2\left|X \cap \overline{Y}\right|}{|X| + |Y|} \tag{7.3}$$

$$FN_{Dice} = \frac{2\left|\overline{X} \cap Y\right|}{|X| + |Y|} \tag{7.4}$$

The Tanimoto similarity measure is defined by:

$$Tan = \frac{|X \cap Y| + \overline{|X \cup Y|}}{|X \cup Y| + \overline{|X \cap Y|}} \tag{7.5}$$

A simple discrepancy measure can be defined based on the area/volume difference between the two regions:

$$AreaDiff. = \frac{||X| - |Y||}{|X| + |Y|} \tag{7.6}$$

Finally, both regions can be compared on a pixel to pixel basis and the difference compared with the average area:

$$PixelwiseDiff. = \frac{X \setminus Y + Y \setminus X}{(|X| + |Y|)/2} \tag{7.7}$$

An important aspect not addressed by the above similarity measures is the distance at which a certain pixel/voxel, not part of the intersection between the two compared regions (X and Y) is positioned regarding the other region. If a pixel x belonging to region X appears at a long distance from region Y, it should count more negatively to the similarity measure than another at a shorter distance. Several similarity measures which incorporate distance can be found in the literature, as described in what follows.

The obtain the Hausdorff distance [66, 67], between two sets of points one starts by computing the minimum distance from each element of X to an element of Y and the largest minimum distance found is chosen:

$$H_{X \to Y} = max(min_{x_i \in X}(\|x_i - y\|)) \tag{7.8}$$

A similar computation is performed for Y towards X, leading to $H_{Y \to X}$. The Hausdorff distance is then given by the larger of the two:

$$H = max(H_{X \to Y}, H_{Y \to X}) \tag{7.9}$$

Other examples of distance based measures are the Yasnoff discrepancy measure [68], defined as:

$$YDM = \frac{1}{N} \sum_{i=1}^{N} d^2(r_i) \tag{7.10}$$

or the factor of merit proposed by Strasters et al. [69]:

$$FOM = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{1 + d^2(r_i)} \tag{7.11}$$

where N is the number of pixels in $X \cup Y$ and $d(r_i)$ is the distance from pixel r_i to the region used for comparison and can be defined as [70]:

$$d(r) = \begin{cases} 0, & r \in X \cap Y \\ min_{x \in X} \|x - r\|, & r \in Y \setminus X \\ min_{y \in Y} \|y - r\|, & r \in X \setminus Y \end{cases} \tag{7.12}$$

Recently, Cárdenes et al. [67] proposed a new measure:

$$JCd = \frac{|X \cap Y|}{|X \cap Y| + \sum_{i=1}^{N} d^2(r)} \tag{7.13}$$

which is similar to Stratsters et al. FOM, presented in eq. 7.11.

All these distance based measures depend on the squared distance from a pixel/voxel r to the closest pixel/voxel in the other region. Consequently, pixels/voxels at a greater distance have a significantly higher contribution to the dissimilarity than those at smaller distances from the reference region.

Goumeidane et al. [71] presented two distortion measures, Internal Distortion Rate (IDR):

$$IDR = \frac{\sqrt{\sum_{r \in Y \setminus X} b_Y^2(r)}}{\sqrt{\sum_{y \in Y}^{N} b_Y^2(y)}} \tag{7.14}$$

which deals with pixels that should have been included in X (because they are present in Y – false negatives, see Fig. 7.1) and the External Distortion Rate (EDR):

$$EDR = \frac{\sqrt{\sum_{r \in X \setminus Y} b_Y^2(r)}}{\sqrt{\sum_{y \in Y} b_Y^2(y)}} \tag{7.15}$$

which accounts for pixels that should not have been included in X (because they are not present in Y – false positives, see Fig. 7.1). In both these measures $b_Y(i)$ is the minimum distance from pixel i to the background (external boundary) of Y.

Working with these two measures allows to distinguish the reason for discrepancy between the two compared regions, as with the positive and negative Dice coefficient measures (eqs. 7.3 and 7.4), while using distance information to balance the measure.

Another approach to discrepancy evaluation, C-Factor, is proposed by Popovic et al. [72]. It is based on a discrepancy measure d_M, defined as:

$$d_M = \frac{2p(1-q)}{p + (1-q)} + \frac{2(1-p)q}{(1-p) + q} \tag{7.16}$$

Table 7.1 – Measures for pixel/voxel set comparison. Type: **S** – similarity: measure is 1 for complete similarity) and **D** – discrepancy: measure is 0 for complete similarity.

Measure	Type	Range	Observations
Jaccard	S	$[0 \dots 1]$	
Dice	S	$[0 \dots 1]$	
FP_{DICE}	D	$[0 \dots 1]$	Disagreement due to false positives (over-segmentation)
FN_{DICE}	D	$[0 \dots 1]$	Disagreement due to false negatives (under-segmentation)
Tanimoto	S	$[0 \dots 1]$	
Volume Diff.	D	$[0 \dots 1]$	Volume disagreement. No info. on overlapping
Voxelwise Diff.	S	$[0 \dots 2]$	Voxelwise comparison
Yasnoff	D	$[0 \dots ?]$	Measure not normalized.
FOM	S	$]0 \dots 1]$	
JCd	S	$]0 \dots 1]$	
IDR	D	$[0 \dots 1]$	Disagreement due to false negatives (under-segmentation)
EDR	D	$[0 \dots 1]$	Disagreement due to false positives (over-segmentation)
C_{FACTOR}	D	$[-1 \dots 1]$	

where

$$p = \frac{TP}{TP + FN} \tag{7.17}$$

$$q = \frac{TN}{TN + FP} \tag{7.18}$$

with TP, the true positive pixels, FP, the false positive, FN, the false negative and TN, the true negative defined as presented in figure 7.1.

The C-Factor is then defined as:

$$C_{FACTOR} = \begin{cases} d_M, & p \geq q \wedge p > 1 - q \\ -d_M, & p < q \wedge p > 1 - q \\ undefined, & p \leq 1 - q \end{cases} \tag{7.19}$$

Parameters p and q, defined above, are usually known as sensitivity and specificity. More information on the meaning of such parameters and their use in slightly different contexts such as receiver operating curves (ROC) and discrete classifiers can be found in [72, 73].

Table 7.1 provides a summary of the presented measures with their variation range and the kind of measure (similarity or discrepancy).

Examples of further studies concerning comparison and analysis of some of the measures described above can be found in Hripcsak et al. [74], Harris et al. [75] and Chang et al. [76].

7.3 Preparatory Study to Choose Similarity measures

To perform the study a data set composed by pairs of LV segmentations was used. The segmentations were obtained using the tool for left ventricle segmentation presented in chapter 5, and included two segmentations (performed by different users or by the same user at different times) for each one of 48 left ventricles in different cardiac phases and belonging to four different patients.

Table 7.2 – Similarity and discrepancy measures used to compare the segmented volumes. The first column refers to the ID used along the text.

#	measure
M1	Straster's FOM
M2	Yasnoff discrepancy
M3	Voxelwise difference
M4	Dice coefficient
M5	False positive Dice
M6	False negative Dice
M7	Jaccard coefficient
M8	Tanimoto's metric
M9	Popovic's C-Factor
M10	Cárdenes JCd

These segmentation pairs are compared both ways, i.e., first segmentation A1 is compared with A2 and then A2 is compared with A1 since a few comparison measures give different results depending on which volume goes first. The set of selected segmentations was chosen to provide, as much as possible, a representative set of the full data to be analysed during the planned evaluation study considering the image data chosen, the cardiac phases and the amount of difference between compared volumes (none, slight and considerable).

Each pair of segmentations was compared using the similarity/discrepancy measures listed in Table 7.2. From the measures described in this chapter (and listed in table 7.1) three were left out of this study: the volume difference, because it does not provide any data on segmentation overlap and the Internal and External Distortion Rates (IDR and EDR). These latter measures are similar to the positive and negative Dice coefficients but use distance to balance the results. Using distance, at least for our particular case, did not yield significantly different results, as depicted in the following section, and therefore IDR and EDR were left out.

The comparison data obtained were then statistically analysed using STATISTICA 6.0[1].

[1]Statistica: `http://www.statsoft.com`

7.4 Results and Discussion

The analysis started with a simple Exploratory Data Analysis [77] (EDA), using boxplots to assess each measure's range and the presence of outliers. Figure 7.2 shows the boxplots for all the measures values, computed for all the cases, where it is possible to notice that M2 extends over a wider range than any of the other measures. This is due to the fact that M2 is not normalized. As the outliers visible for M2 could be due to some numerical error (as they are just a few and really stand out) they were temporarily removed from the set in order to test if they influenced the results that follow (see how M2 appears isolated in the dendogram of figure 7.4) but no notable change was observed and thus the original set was kept.

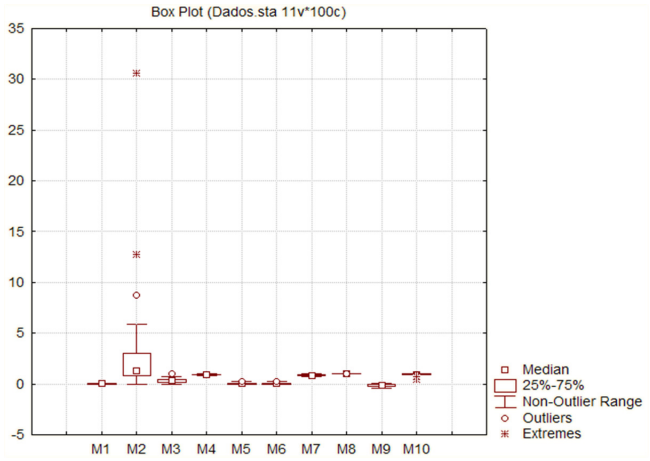


Figure 7.2 – Boxplots for all the used similarity and discrepancy measures.

One important analysis which can be performed is to assess if, from the set of cases (i.e., pairs of volumes being compared), there is any which stands out. This can be important to detect cases that have particular characteristics which one might be interested in removing from the evaluation, or in adding more similar cases if it is recognized as important for the evaluation. It might also reveal some lack of robustness of the measures in dealing with a particular situation.

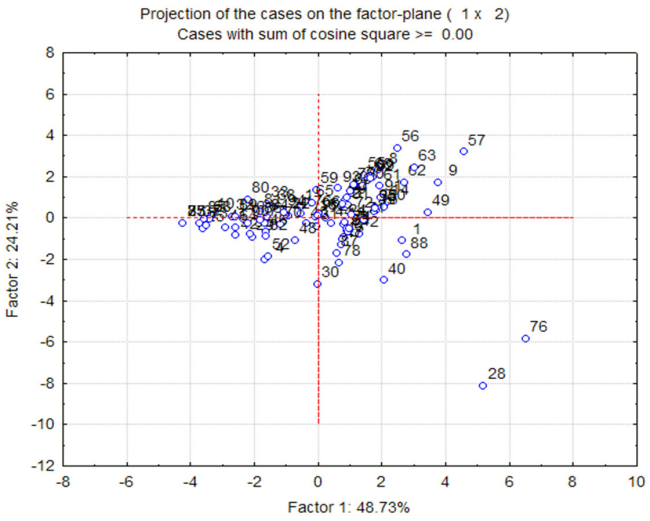


Figure 7.3 – Factorial plane depicting all 96 cases.

Figure 7.3 shows a factorial plane depicting all 96 cases analysed (48 pairs compared both ways). Notice that one of the pairs (pair 28 and its inverted version, 76) is clearly apart from the remaining pairs. This might be due to an error in the data or some particular characteristic not found in other pairs. Pair 28 was responsible

for the highest outlier in M2 and a closer look into the volumes revealed a badly stored volume (on disk) which resulted in a discrepancy significantly higher than that found for other pairs.

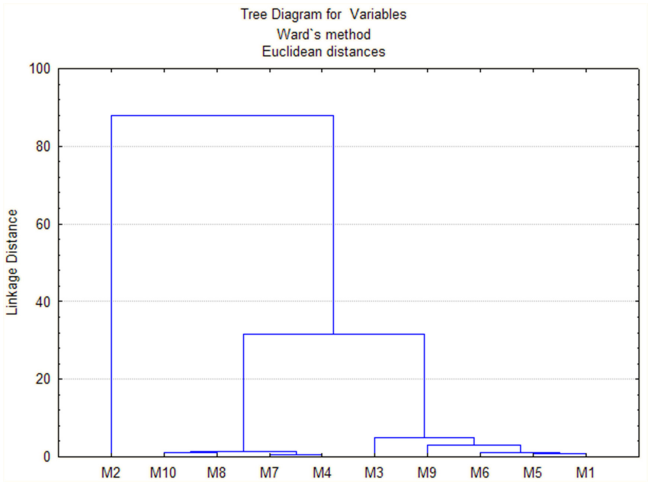


Figure 7.4 – Dendogram depicting the association between measures.

To study the association between measures in order to find possible redundancies in the results they provide, Cluster Analysis [78] was used. Looking at the dendogram presented in figure 7.4 (using Ward's method as a measure of proximity) it is possible to detect several associations between measures: M1, M3, M5, M6 and M9 form a group; M4, M7, M8 and M10 form another group; and M2 stands by itself and clearly apart from the remaining measures. This means that the used measures basically behave in three different ways with considerable redundancy being observed.

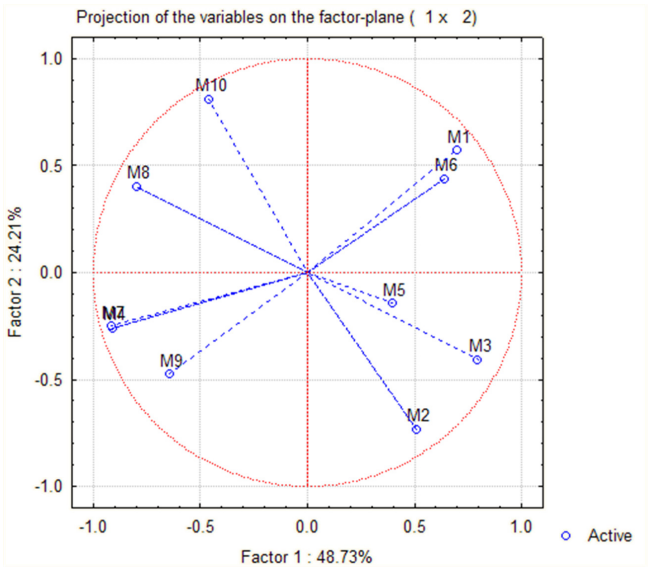


Figure 7.5 – Factorial plane showing the principal components for each similarity measure.

Figure 7.5 shows the factorial plane depicting the principal components associated with each measure. Notice that M4 and M7 completely overlap (see third quadrant). This is, in fact, expected as these two measures, the Dice coefficient (M4) and the Jaccard coefficient (M7) are known to be equivalent. This detail also serves as a confirmation that the analysis is yielding coherent results.

Figure 7.6 shows a matrix containing plots of each measure versus the remaining measures which can be used to get additional insight on how the measures compare to each other. For example, notice how measures M2 and M10 result in plots with points concentrated to one

Figure 7.6 – Matrix showing plots of each measure versus the remaining measures.

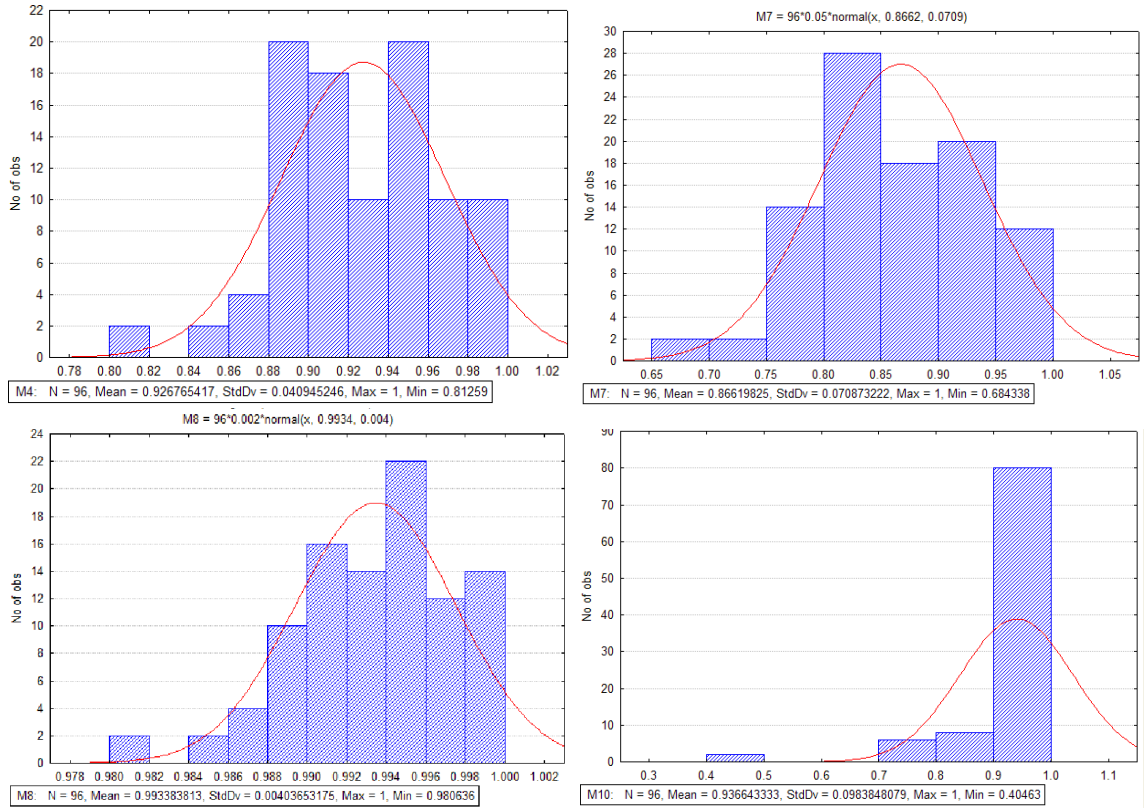


Figure 7.7 – Histograms showing value distributions for measures in the first group: M4, M7, M8 and M10.

side. This can be explained since these are the only two measures which inversely depend on the squared distance from a voxel to the reference volume (see equations 7.11 and 7.13). Notice also that the points are grouped at different sides for each measure: one is a similarity measure and the other a discrepancy measure.

Having identified the three main groups of measures it is possible to perform the analysis using just three measures instead of ten. At this point it is necessary to choose a measure which represents each group. This can be done by using several criteria such as computational cost or discrimination capacity. For example, for the latter criterion consider figure 7.7 where the histograms for the data provided by measures M4, M7, M8 and M10 (grouped in the dendogram of figure 7.4) are presented. Notice that for M7 the range of values obtained shows considerable distribution of observations along the different bins while in M10 there are only four classes with observations (and empty classes in-between) with one of them containing most observations on the upper part of the range. This shows that, potentially, M7 might yield better discrimination and thus be the better measure of the two.

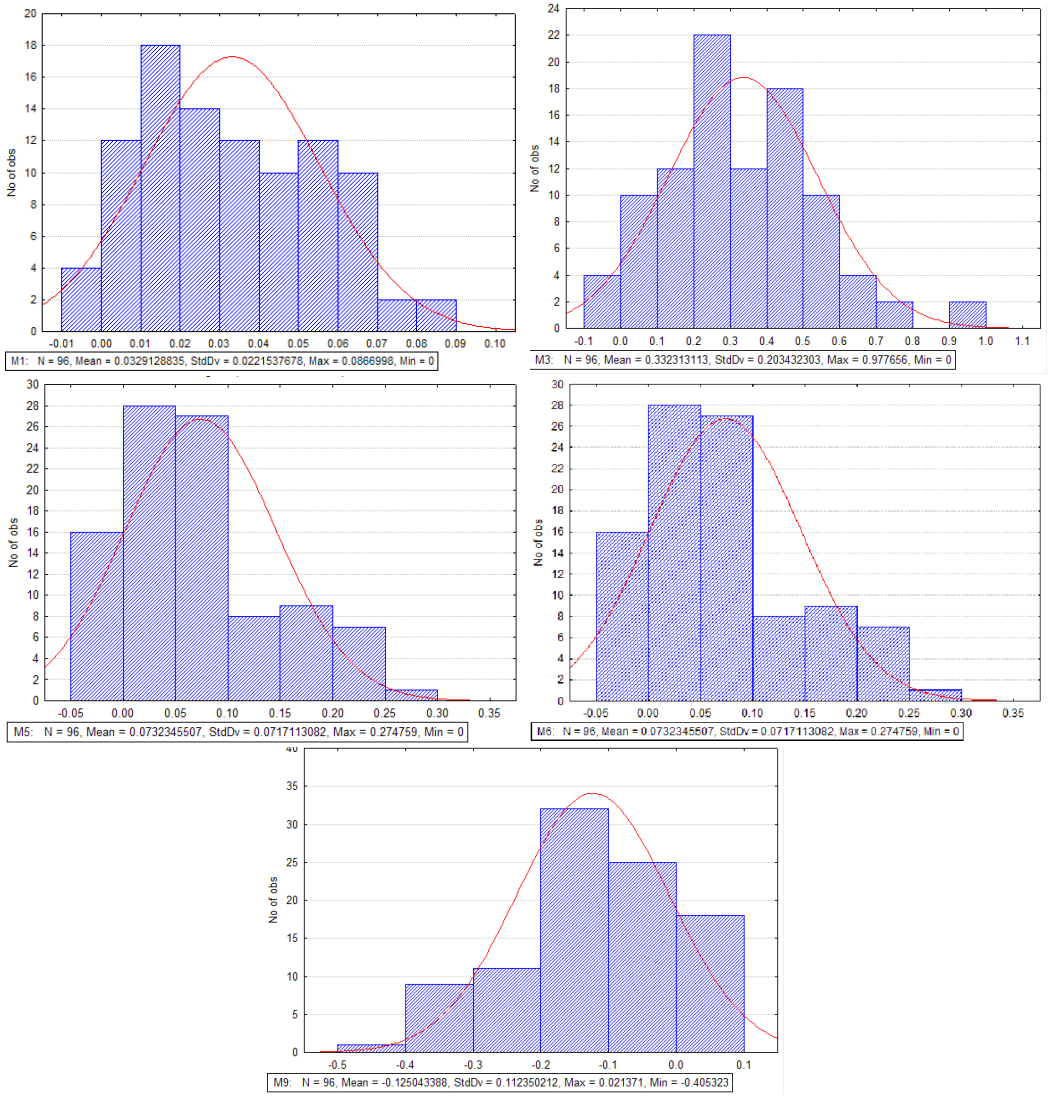


Figure 7.8 – Histograms showing value distributions for measures in the second group: M1, M3, M5, M6 and M9.

Figure 7.8 shows the histograms[2] for the data provided by metrics M1, M3, M5, M6 and M9 (grouped in the dendogram presented in figure 7.4).

Regarding measure symmetry, i.e., their behaviour when computed both ways for the same pair of volumes, the 96 comparison cases (48 pairs of volumes compared both ways) have been split into four groups (to improve visibility when depicting the results) of 24 cases each. Notice

[2]The histograms for measures M1, M3, M5 and M6 were built defining each bin with an exclusive (open) lower boundary and an inclusive (closed) upper boundary [79]. As a result, even though these measures vary between zero and one, the histograms present a bin with a negative lower boundary. This is due to the fact that the value 0 (zero) occurred several times for each metric and could not be included in the bins starting in zero since it has an open boundary. Therefore, the presence of these leftmost bins (with negative lower boundaries) should no be interpreted as meaning the occurrence of negative values.

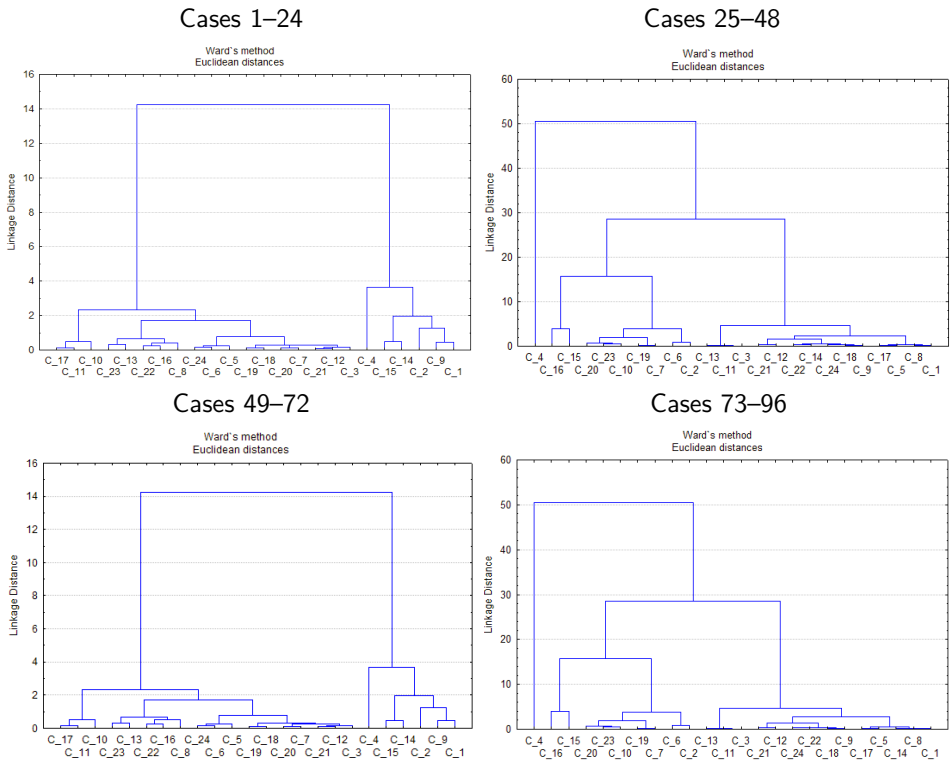


Figure 7.9 – Dendograms showing, on the left, grouping for cases 1–24 and 49–72 and, on the right, for cases 25–48 and 73–96. These dendograms provide evidence that, with the tested measures, the order of comparison does not influence the final result.

that cases 49–96 correspond exactly to cases 1–48 but with the volume order changed for comparison. Figure 7.9 shows dendograms for cases 1–24, 25–48, 49–72 and 73–96. Notice that the dendograms on the left are very similar to each other and the same happens to the dendograms on the right. This shows evidence that no matter what is the order of comparison, with this set of measures, differences/similarities between cases are depicted likewise. This, however, does not mean that both orders can be used interchangeably.

Therefore, regarding measure grouping, complexity, range and discrimination, we consider measures M3 (voxelwise difference), M7 (Jaccard coefficient) and M2 (Yasnoff discrepancy) appropriate for comparing LV segmentations.

It is also important to notice that the Yasnoff discrepancy is not normalized, which might be the main reason why it behaved so differently from the remaining measures. Straster's FOM, for example, which also depends on the squared distance to the reference region (see eqs. 7.10 and 7.11), was grouped with other measures. Furthermore, a non-normalized measure is difficult to interpret. Therefore, during the evaluation study presented in chapter 8 we did not consider the Yasnoff discrepancy.

7.5 Conclusions

A methodology is presented which allows selecting similarity measures to compare left-ventricle segmentations. This preparatory study uses a small set of segmentation pairs (when compared with the final set), to test several similarity measures described in the literature and then select those which yield relevant results, based on redundancies and discrimination capacity assessed by a statistical data analysis. This way, the complete comparison study can concentrate on the relevant measures, reducing the amount of data generated for analysis.

In our particular scenario, concerning the comparison of LV segmentations, we selected measures M3 (voxelwise difference) and M7 (Jaccard coefficient) based on several criteria.

To the best of our knowledge this kind of preparatory study is not common in the segmentation validation studies literature. In fact, most studies tend to use a few measures just because they are commonly used without assessing if they are fit for their particular case or if additional measures really might bring additional insight.

Using this methodology it was possible to reduce the number of used similarity measures from ten to three (one for each group found) while choosing, for example, measures which are expected to have good discrimination properties.

We emphasize that even though the methodology presented in this chapter is applied to a particular dataset of LV segmentations it is general enough to be applied in similar situations where the segmented regions and measures might be different.

Chapter 8

Left Ventricle Segmentation Tool Evaluation

"The only man who behaves sensibly is my tailor; he takes my measurements anew every time he sees me, while all the rest go on with their old measurements and expect me to fit them."

– George Bernard Shaw

USING any kind of segmentation tool to obtain data to guide diagnosis or perform quantitative analysis requires a proper assessment of its performance. Regarding the LV segmentation tool presented in chapter 5, complemented with the 3D editing tool described in chapter 6, an evaluation study has been performed in order to assess segmentation time, intra and inter-observer variability and user satisfaction.

Results show that the presented segmentation method allows segmentation in reasonable time with small intra- and inter-observer variability.

Publications:

The work presented in this chapter has been partially published in:

- Samuel Silva, Joaquim Madeira, Beatriz Sousa Santos, "Inter-observer Variability Assessment of a Left Ventricle Segmentation Tool Applied to 4D MDCT Images of the Heart", Proc. 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society – EMBC 2011, pp. 3411–3414, Boston, USA, September, 2011

8.1 Introduction

On the preliminary qualitative evaluation, described in chapter 5, the results provided by the proposed LV segmentation algorithm have been assessed. The outcomes showed that the method provides good segmentations with only some minor problems occurring in identifying the mitral valve plane or correctly segmenting the septal section of the myocardium. To deal with those problems a method was devised (and implemented in CardioAnalyzer, as described in chapter 5) that includes the segmentation algorithm and a guided procedure which walks the user throughout the revision (and editing [80], if necessary) of the obtained segmentations.

Since the segmentations performed in CardioAnalyzer are supervised by radiographers our main concern is to assess intra and inter-observer variability.

For that purpose a quantitative evaluation has been carried out involving three radiographers which segmented LVs in a set of four multiple detector-row computed tomography (MDCT) coronary angiography exams, using CardioAnalyzer. The segmentations were then compared using the Jaccard coefficient and voxelwise difference, chosen using the method described in chapter 7.

Results show that the segmentation tool allowed segmenting the LV with low intra- and inter-observer variability in a short time.

This chapter is organized as follows. Section 8.2 presents a brief overview on notable works and guidelines regarding medical image processing methods evaluation, as well as several agreement assessment metrics used for segmentation comparison. Section 8.3 presents the methodology used to perform the evaluation, describing the different aspects of the experimental protocol adopted. Section 8.4 presents the outcomes of the analysis performed over the collected data. Finally, some conclusions are presented in section 8.5.

8.2 Background Work

Evaluation of image processing methods is of paramount importance. To support the work presented in this chapter a brief overview of notable principles and applications concerning the evaluation of segmentations methods is provided.

8.2.1 Evaluation of Image Processing Methods

Zhang [81, 82] classifies segmentation evaluation methods in two main categories: analytical and empirical. The analytical methods evaluate the segmentation by looking into the segmentation algorithm and analysing its properties. On the other hand, the empirical methods evaluate the algorithms by applying them to test images and then assessing the quality of the results. In this category two types of evaluation methods are identified: those which use goodness criteria

(also known as unsupervised methods; a recent survey is available in [83]) and those which use discrepancy criteria. The goodness measures are often defined according to intuition and concern aspects such as region homogeneity, region contrast and region shape [82]. The discrepancy measures account for the difference between a segmentation and a ground truth, i.e., an ideal segmentation considering features such as the number of misclassified pixels. Several comparative studies have shown that discrepancy methods are more effective than goodness methods [82] but some advantages might arise in the latter due to their unsupervised nature [83].

Zhang also considers some particular evaluation methods which do not clearly fit into any of the previously mentioned categories. One such example is when, after running the segmentation algorithm, the image needs to be manually edited to correct the results. The amount of editing and time taken can then be used to provide a measure on how the initial segmentation differs from the desired result. Such a method, being an evaluation focused on a particular task (editing), has a direct dependence on the editing tools available. More accurately, it can be argued that such is an evaluation of the post-processing task rather than an evaluation of the segmentation method.

When performing an evaluation, and designing the experimental protocol to be used, it is common that a lack of proper definition of the context, input data and tasks, as well as an incomplete description of the experimental procedures and analysis methods, hinders drawing strong conclusions concerning the performance of the methods being evaluated and a comparison with other evaluations.

Buvat et al. [84] propose a set of rules to guide the design and application of evaluation protocols for medical imaging processing methods.

Table 8.1 – Possible evaluation levels in medical imaging procedure assessment according to Buvat et al. [84].

Level	Evaluated aspect
1	Technical efficacy
2	Diagnostic accuracy efficacy
3	Diagnostic thinking efficacy
4	Therapeutic efficacy
5	Patient outcome efficacy
6	Societal efficacy

Considering a particular level of evaluation (ranging from technical efficacy up to the social impact of the method in the society; see table 8.1) and considering the method to be evaluated, the general purpose of the evaluation (abstract aim) study should be defined. Pursuing the general purpose requires that a context, i.e., the environment in which the experimental protocol will be designed, is established.

Projecting the initial general purpose in a particular context (see figure 8.1) leads to a hypothesis which will be tested during the evaluation process. Nevertheless, fulfilling the initial purpose might require different approaches (contexts) to be considered. Therefore, sometimes more than

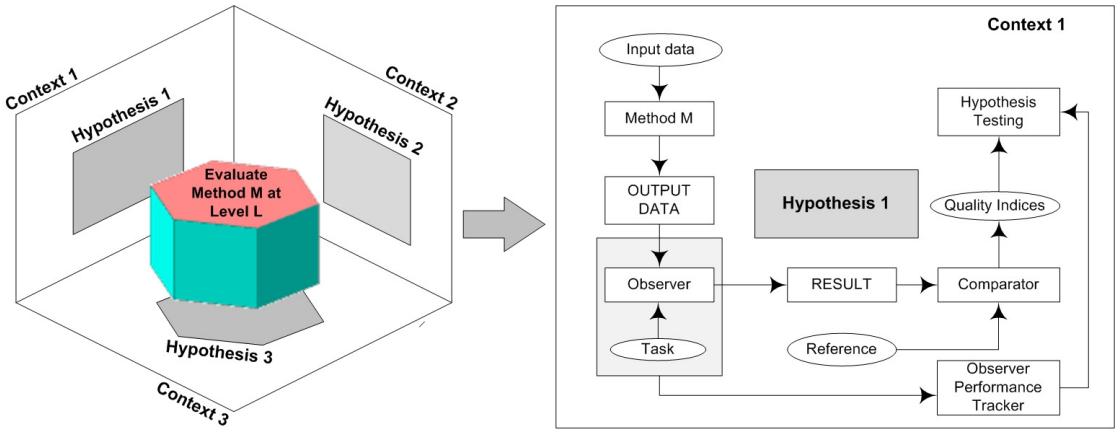


Figure 8.1 – Evaluating a method at a specific level can be performed in different contexts. Choosing one context leads to an hypothesis which then determines other components in the evaluation, such as the input data or how observer and method performance assessment is carried out. Figure adapted from Buvat et al. [84].

one evaluation study must be performed [84].

The hypothesis will then determine the type of input data that should be used and the task that will be performed by the observer over the outputs of the method in order to obtain proper results to be used to assess the method's value. How this is performed is closely related with the hypothesis.

Finally, data from the quality indices and any additional data obtained, for example, by monitoring observer performance, is used to test the hypothesis and determine if it is rejected or if there is not enough evidence to support it.

The various elements involved in an evaluation and their dependences are depicted in figure 8.1. It is important to notice that these elements must be defined as completely as possible. For instance, it must be stated if the input data is real or synthetic, how it was chosen and a proper size of the input data set must be determined; the task must be precisely defined and the observer's profile characterized by stating his/her level of expertise, which information will be provided and presenting an assessment of the intra and inter observer variability; the quality indices must be appropriate to test the hypothesis and, if possible, more than one should be used [84, 67]; and the hypothesis test should use methods compatible with the properties of the quality indices chosen. Another important factor is to ensure that all observers perform the tasks in similar conditions, e.g., similar computers, display contrast/ brightness and ambient light.

Udupa et al. [85] identify three main factors for evaluation of image segmentation methods: precision, accuracy and efficiency. Assessing precision (reliability) includes using a set of quality indices and then performing a statistical analysis of their variation. Assessing accuracy requires a

ground truth or, since it is often impossible to obtain it, a surrogate of truth which will be used as reference in a procedure similar to that used to evaluate precision. In evaluating these two factors, the comparison measures chosen will be limited by the characteristics of the segmentation data and used reference. Efficiency assessment consists in evaluating the computational and user time required for training and performing the segmentation.

8.2.2 Agreement indices

The similarity measures presented in chapter 7 allow to compare each user's segmentation with all the corresponding segmentations from other users. Since we do not have a unique ground truth (instead, all segmentations are ground truths as they are provided by qualified observers) it can be interesting to test the agreement of each user against a group formed by the remaining users.

The usual and intuitive method is to determine a consensus among all raters and then compare each one individually with the consensus [86]. Nevertheless, the method by which this consensus is obtained is not trivial and several have been discussed in the literature (Vanbelle et al. [86] present a brief overview), which might even lead to different conclusions about the same data.

The Williams index is an agreement measure widely used in the literature [87] [66], which can be expressed as:

$$WI_i = \frac{(n-2)\sum_{j\neq i}^{n}\delta_{ij}}{2\sum_{j\neq i}^{n}\sum_{k\neq i}^{j-1}\delta_{jk}} \tag{8.1}$$

where δ_{ij} is the similarity/dissimilarity value between segmentation i and segmentation j, obtained by using one of the measures described in chapter 7 (e.g., Jaccard coefficient), and n is the number of raters (segmentations). It avoids determining the consensus [86] by comparing the mean agreement between one rater and each of the remaining raters with the mean agreement between all possible pairs in the group.

William's index values can belong to three notable ranges [66] as presented in table 8.2.

STAPLE (Simultaneous Truth and Performance Level Estimation) is an agreement index presented by Warfield et al. [88]. This method allows computing a probabilistic estimate of the true segmentation based on a set of segmentations obtained from different raters (human or computer). It also computes a performance level for each one of those segmentations.

The probabilistic estimate of the true segmentation is obtained by computing an optimal combination of the segmentations based upon their performance level, a prior model for the

spatial distribution of the structures being segmented and some spatial homogeneity constraints. This optimal combination can then be considered as the ground truth and the metrics described in chapter 7 can be used to assess each segmentation (see Martin-Fernandez et al. [89]). Given the numerous aspects involved in this method the reader is referred to Warfield et al. [88] for a detailed description.

As reported by Martin-Fernandez et al. [89] the results obtained using William's index and STAPLE are quite similar and unless a ground truth is absolutely necessary they advise using William's index since its computation is much faster.

Table 8.2 – The William's index can provide a value in three notable ranges which lead to different interpretations.

$WI_i > 1$	The remaining segmentations are more similar to segmentation i than to each other
$WI_i \approx 1$	The remaining segmentations are as similar to segmentation i as they are to each other
$WI_i < 1$	The remaining segmentations are more similar to each other than each is to segmentation i

Bouix et al. [87] use multidimensional scaling (MDS) to explore similarity data among segmentations. This method allows the representation of similarity measures among pairs of segmentations as distances between points in a lower dimensional space (in this particular case, 2D). It takes the similarity measurements for all possible segmentation pairs, computed using one of the metrics described in chapter 7 and it tries to find the optimal disposition of all points in 2D space (one for each segmentation). This is performed in such a way as to preserve the similarity data among segmentations but now expressed as a distance in 2D space (in general, the Euclidean distance). This is accomplished by minimizing a function of the error between the distances given by the similarity measurements (δ_{ij}) and the Euclidean distance in 2D space (d_{ij}):

$$E = \frac{1}{c} \sum_{i=1}^{n} \sum_{j=i}^{n} \left[\frac{(\delta_{ij} - d_{ij})^2}{\delta_{ij}} \right] \tag{8.2}$$

where

$$c = \sum_{i=1}^{n} \sum_{j=i}^{n} \delta_{ij} \tag{8.3}$$

The measures described above allow computing the agreement for each segmentation towards a reference obtained from the remaining segmentations (or towards each other). Another option can be to compute a metric which, in a single value, expresses the conformity among all the segmentations (more than two). This issue has already been addressed by several authors. A first approach (an example can be found in Hurkmans et al. [90]) is to obtain a conformity index

based on the ratio between the region resulting from the intersection of all segmentations and the region resulting from the reunion of all segmentations.

$$CI_{common} = \frac{|X_1 \cap X_2 \cap \ldots \cap X_n|}{|X_1 \cup X_2 \cup \ldots \cup X_n|} \tag{8.4}$$

This is a simple generalization of the Jaccard coefficient for multiple segmentations.

Another approach can be to first compute a similarity measure for all possible segmentation pairs (e.g., the Jaccard coefficient) and then compute the average value.

As pointed by Kouwenhoven et al. [91], a disadvantage of CI_{common} is that it is strongly dependent on the number of segmentations considered which can be an obstacle to, e.g., a comparison between studies which use a different number of segmentations. To cope with this issue, Kouwenhoven et al. propose a generalized conformity index, CI_{gen},

$$CI_{gen} = \frac{\sum_{i=1}^{n} \sum_{j=i+1}^{n} |X_i \cap X_j|}{\sum_{i=1}^{n} \sum_{j=i+1}^{n} |X_i \cup X_j|} \tag{8.5}$$

which expresses, in a single value, the conformity among a set of segmentations without a dependence on the number of segmentations. If there is no overlap among the different segmentations CI_{gen} is equal to 0. On the other hand, if all segmentations completely coincide CI_{gen} is 1. As stated earlier, this generalized conformity index is different from the agreement indices above since it does not provide an individual value for each segmentation but a global characterization of the segmentation set.

8.2.3 Segmentation Evaluation Examples

In this section we perform a brief overview of recent evaluation methods presented in the literature and focusing on assessing the performance of medical image processing procedures. The purpose is to highlight the main features of the described studies which are relevant for our work.

Coche et al. [92] compare an automatic and a semi-automatic segmentation method for quantitative left and right ventricular global functional analysis. Both methods are compared based on the end-systolic and end-diastolic volumes, the stroke volume and the ejection fraction for both ventricles. The evaluation was performed over a set of 15 exams from patients suspected of having a pulmonary embolism and half of them had cardiac or lung diseases. Intra-observer variation (reproducibility) was also assessed by asking one of the observers to repeat the evaluation one month later.

Babalola et al. [66] perform the evaluation of four methods for segmenting different brain structures from magnetic resonance (MR) images. The evaluation uses a set of 270 images which were manually segmented to identify the structures of interest and were used as the gold standard. The intra and inter-observer variability for the manual segmentation method had already been evaluated in a previous study. Then, the four automatic segmentation methods were applied to the images and the results compared with the gold standard using several metrics including the Dice coefficient, the Jaccard coefficient, the false positive and false negative Dice coefficients, the Hausdorff distance and volume similarity. Even though the manually segmented images are considered the ground truth this is, in fact, an assumption since the ground truth is unknown. To evaluate the agreement among all the segmentation methods and the assumed gold standard the William's index was used.

To reduce observer variation in tumour delineation in lung cancer, Steenbakkers et al. [93] performed an evaluation study in two stages. In the first stage, the variation associated with the current delineation method was evaluated: eleven radiation oncologists delineated tumours in CT lung scans from 22 patients. This first phase was also useful to obtain data which allowed optimization of the delineation software and delineation protocol (i.e., the steps and rules followed to perform a delineation). The improved software and protocol were tested after a year, in the second stage of this study: the same eleven radiation oncologists were asked to delineate tumours for the same 22 patients, using both CT and 2-[^{18}F]fluoro-2-deoxy-D-glucose positron emission tomography (FDG-PET) exams. Since the study was performed in several institutions care was taken to ensure that all oncologists delineated the tumours in identical conditions: similar computers and monitors were used and contrast and brightness was calibrated among them using a test exam. An interesting aspect of this study is that during the delineations, in both stages, all interactions performed by the oncologists were recorded in order to account for the delineation time, number of delineation points (dominant points of the delineation), number of corrections and respective region [94]. The analysis was performed, for each patient, comparing the common volume, i.e., the volume common to all delineations with the encompassing volume, i.e., the minimum volume which encompasses all delineations (which is expressed by metric CI_{common} presented in eq. 8.4).

8.3 Experimental Protocol

The following sections provide details regarding the devised experimental protocol. As previously explained, the main purpose is to assess if the devised segmentation method and proposed 3D editing tool allow obtaining LV segmentations with a small intra-observer and inter-observer variability. It is also important to understand how long does the segmentation take and how

much editing has to be performed in order to obtain a segmentation judged adequate by the radiographers.

8.3.1 Subjects

Three experienced radiographers (from now on generically referred to as RADIOGRAPHER A, RADIOGRAPHER B and RADIOGRAPHER C) who have everyday contact with acquiring, segmenting and analysing coronary CT angiography (CTA) images have participated in this evaluation.

8.3.2 Test Data Sets

Four exams have been chosen from the set of exams performed during one week time at Centro Hospitalar de Vila Nova de Gaia/Espinho (CHVNG/E). Care has been taken not to include exams which presented any serious acquisition artefact, but that was the only rejection criterion used.

Since most of the cardiac phases available are similar (diastolic stage of the cardiac cycle) and in order not to extend the evaluation time, overloading users with a large number of segmentations per exam, 6 cardiac phases were selected: the reference phase (60%), the end-systole (typically at 25%) and its neighbor phases (15% and 35%) which are usually significantly different and, finally, the end-diastole (typically at 95%) and a last phase, midway between the reference and end-diastole (75%).

8.3.3 Protocol

All radiographers were asked to use CardioAnalyzer to segment the endocardium and epicardium for all the exams and selected phases. All CardioAnalyzer features which did not concern segmentation were disabled (e.g., LV functional analysis) in order to avoid distractions, thus allowing the users to focus on the relevant task. CardioAnalyzer would present an exam to the user, when started, and would exit after the segmentation was performed.

The order in which the exams were presented to each user was randomly selected to avoid any effect related to the sequence in which exams were segmented.

For intra-observer variability evaluation each radiographer had to perform each exam's segmentation twice, at different times, reasonably set apart, to avoid memory effects. The second segmentations were performed seven months after all the first segmentations had been accomplished and, again, in a random order.

User Training and Protocol Validation

Since the evaluation was to be performed without direct supervision, to allow radiographers to perform the evaluation tasks at the time of their choice (e.g., in-between their work tasks), a preliminary session was scheduled to present the task and a small "user guide" was provided to support them along the evaluation.

All radiographers received an introductory explanation about the whole evaluation process and started the evaluation with a training exam (not accounted for during analysis). This allowed the radiographers to get acquainted with the different steps of the segmentation process and the tasks they had to perform. After being given some time to test the segmentation tool using the test exam they were allowed to make any comment/question and the beginning of the "real" evaluation was left at their discretion (when they felt confident using the segmentation tool).

Given the possible influence of ambient conditions, such as lighting and computer screen resolution and brightness, all radiographers performed the segmentation in the same computer and location (where they perform their everyday segmentation and analysis tasks).

8.3.4 Collected Data

During the segmentation procedure, for each radiographer and each exam, the following data were collected:

- Proposed and accepted mitral valve level
- Proposed first segmentation for all cardiac phases
- Accepted segmentation for all cardiac phases (possibly including editing using the 3D tool presented in chapter 6)
- Time taken to approve each cardiac phase
- Overall evaluation time, also including computation times

After concluding the segmentation for all exams, the radiographers were invited to state their opinions about CardioAnalyzer as a tool for left ventricle segmentation and asked how they would compare it with the software packages they use to perform LV segmentation, namely the Argus and Circulation packages by Siemens and the Aquarius workstation by TeraRecon.

8.4 Evaluation Study Results

Even though comparison has been performed using both the Jaccard coefficient and voxelwise difference (according to the conclusions presented in chapter 7), for the sake of simplicity only data for the first is presented given that, for the performed analysis, no additional relevant insight or conclusions were obtained from the voxelwise comparison.

8.4.1 Segmentation Time

Looking into the data collected during the evaluation some extraneous values were detected. Since this evaluation was performed by the radiographers while they were at work, it might occur that they had to leave the evaluation at any point to attend to another matter. This resulted in interaction times that did not reflect the true duration of the evaluation: for example, in one instance, an endocardium segmentation had a completion time of around 29000 seconds. This is an extreme example and not many extraneous values were found. For the first time the evaluation was performed we found seven awkwardly high times and for the second run (for intra-observer variability assessment) we found four.

To deal with these extraneous times the following methodology was used: the average time to perform an endocardium/epicardium segmentation was computed leaving these values out. Next, in order to be able to compute total times per exam and total evaluation times the extraneous values were replaced with the average times found. This is a different approach than that presented in Silva et al. [95], which did not deal with these extraneous values.

The average times to perform different tasks during both runs of the evaluation study (denoted e1 and e2) are presented in table 8.3.

The first notable aspect is that times obtained during the second evaluation run are generally lower than those obtained in the first run. This might be the effect of improved familiarity with the segmentation tool and tasks, given the experience obtained during the first evaluation run. This is also a good indication that further training might improve performance. In the discussion that follows the considered times are those of the second evaluation run.

Notice that the average time needed to perform the segmentation of six cardiac phases (1 exam) was less than 13 minutes and included segmenting 6 endocardia and 6 epicardia. This is a very good time when compared to semi-automatic segmentation times reported in the literature. For instance, Coche et al. [92] report 15-20 minutes to segment the endocardia and epicardia for two cardiac phases (end-systolic and end-diastolic) using a semi-automatic method. Muhlenbruch et al. [32] report around 170 seconds to segment the endocardium for two cardiac phases which is a similar time to the one obtained using our method for segmenting, for example, a cardiac

phase (156 s, endocardium + epicardium) or two endocardia (140 s).

The total interaction time refers to the average time taken by the radiographer interacting with the segmentation method (revising and editing the segmentation) while the evaluation time concerns the average time for the whole evaluation study (including loading images from disk, processing and answering the questions at the end of each exam segmentation). Considering the total evaluation time, and considering that 24 segmentations were performed, one can estimate an average time around three and a half minutes to perform each segmentation (already including processing and image loading times).

Table 8.3 – Average time, standard deviation and median (over the three subjects) to perform different tasks during both runs of the evaluation study (e1 and e2).

	avg. time (s)		**std. dev.**		**median (s)**	
	e1	e2	e1	e2	e1	e2
endocardium	70	74	65	107	51	37
epicardium	85	53	90	51	74	39
cardiac phase	156	127	116	122	126	103
exam	934	764	377	434	853	711
interaction time	3734	3055	1178	1049	4044	2682
evaluation time	5141	4911	1328	656	5423	4915

8.4.2 Amount of Performed Editing

The segmentation method provides a set of high level parameters (e.g., mitral valve plane) that can be tuned to obtain the best possible segmentation without editing. After that first version, the user can perform editing where necessary. A first analysis concerned the difference between the first version, provided by the segmentation method, and the final version, approved by the radiographer, possibly after editing.

Figure 8.2 shows values obtained with the Jaccard coefficient for the comparison between these two versions for all cardiac phases and exams. Notice that, for all phases, both segmentations are very close to each other, with the Jaccard coefficient above 0.85 (exception for the 60% cardiac phase in Exam 1).

Notice that the 60% phase is, consistently, the one presenting a larger difference between the first proposed segmentation and the one approved by the radiographer. This is to be expected since it is the reference phase used to guide the segmentation of the remaining phases. Corrections in this phase will improve the output of the remaining phases.

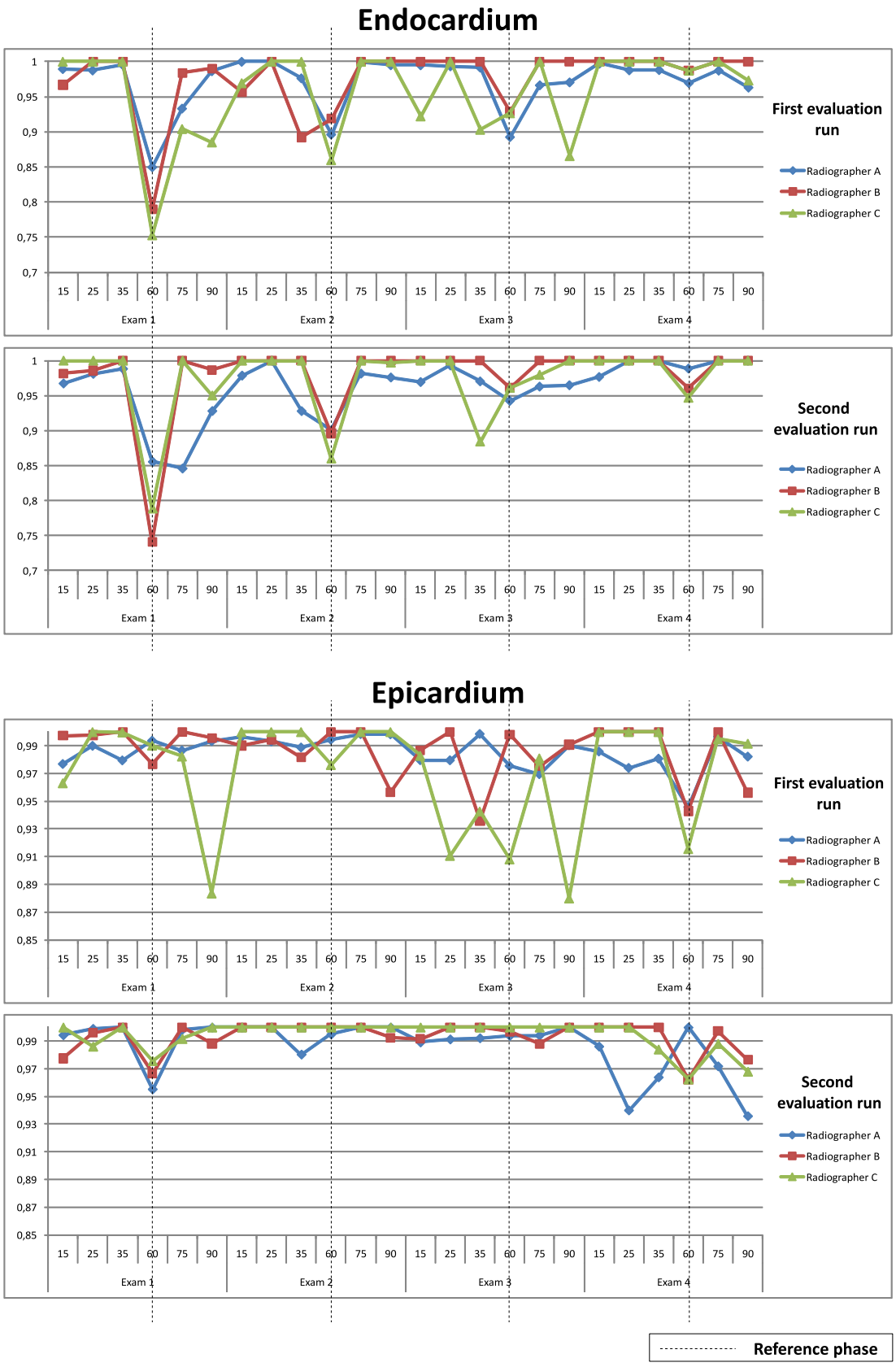


Figure 8.2 – Amount of editing performed: Jaccard coefficient values comparing the segmentation provided by the segmentation method and the corresponding segmentation after radiographer approval (possibly after editing).

8.4.3 Intra-observer Variability

Considering the segmentations performed by the radiographers at two different times (separated by seven months) it is possible to assess intra-observer variability.

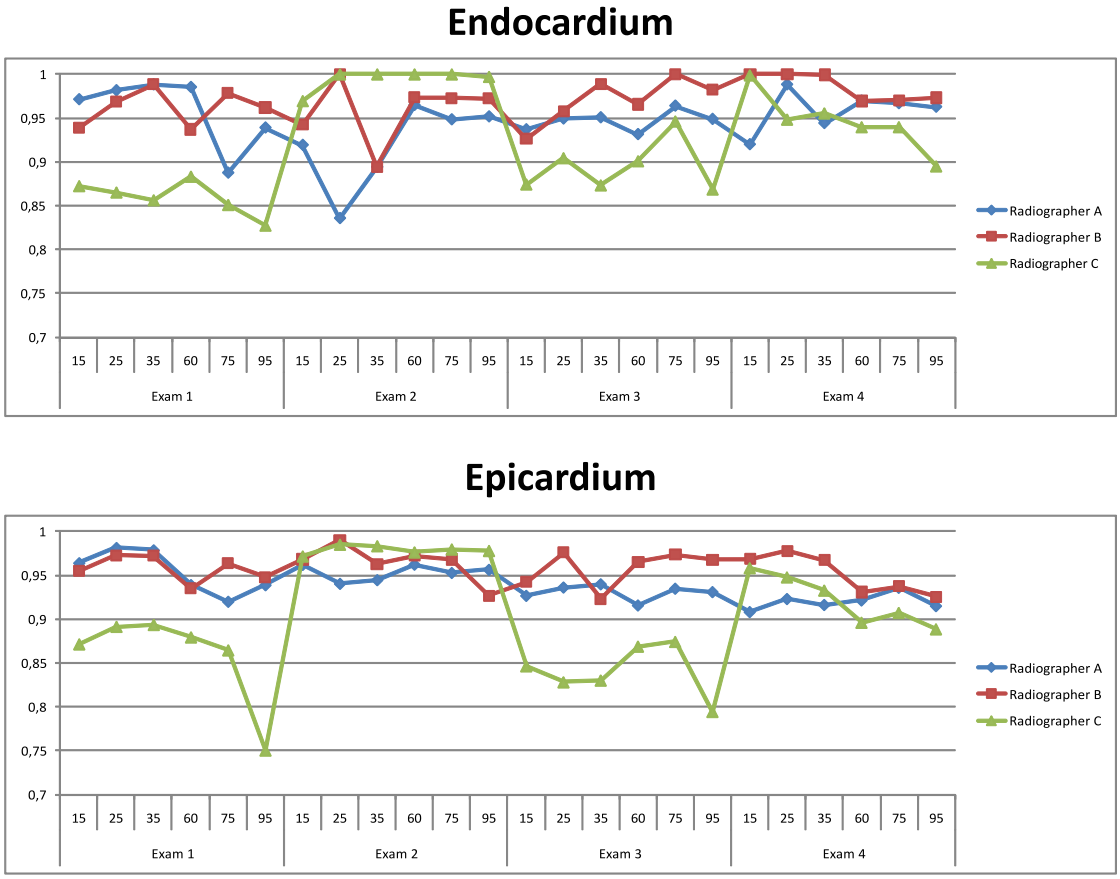


Figure 8.3 – Intra-radiographer variability using the Jaccard coefficient: each radiographer segmentations for both evaluation runs were compared.

Figure 8.3 presents comparison data for corresponding segmentations performed by each radiographer during the two evaluation runs. Notice that, for both endocardium and epicardium, values are generally above 0.90 (with 1.0 meaning complete match) with Radigrapher C presenting the largest differences.

8.4.4 Inter-observer Variability

The results obtained using the Jaccard similarity coefficient (figure 8.4) show evidence (values ≥ 0.85; with 1 meaning complete match) that for each radiographer the segmentations have a high degree of similarity when compared individually with the corresponding segmentations

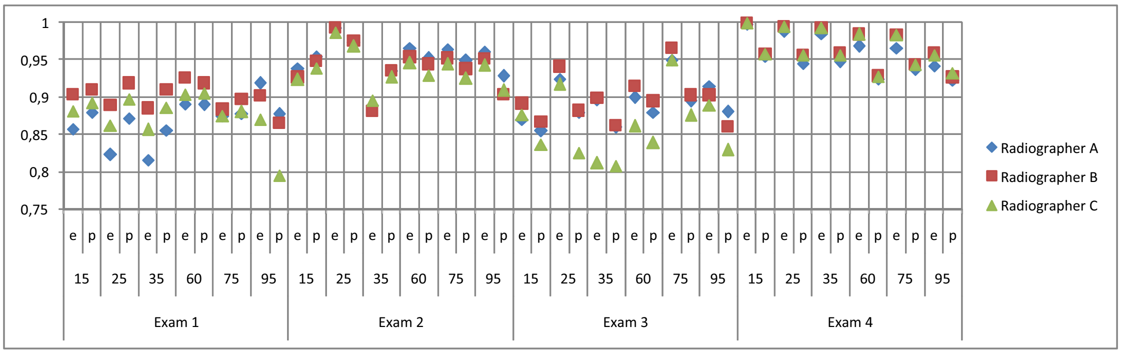


Figure 8.4 – Inter-radiographer variabilty: mean Jaccard similarity coefficient value obtained by the radiographers for each cardiac phase (e – endocardium; p – epicardium) and exam.

performed by the remaining radiographers.

The Williams' agreement index (figure 8.5) provides additional data concerning the segmentations as it presents how each segmentation can be compared with the set composed of the remaining two. Overall, the results show good agreement (with the Williams' index close or above 1) for both evaluation runs. Nevertheless, the Williams index showed some poorer results (bellow 0.90) for some of the segmentations of Radiographer C (see figure 8.5).

Finally, all segmentations performed during both evaluation runs were gathered and corresponding segmentations cross-compared irrespective of the evaluation run they were obtained from. The main goal was to obtain a global idea of the whole evaluation process.

Figure 8.6 shows agreement values for each radiographer and considering the segmentations obtained for both evaluation runs. Notice that, apart from Radiographer C, during the first evaluation run, remaining values are mostly higher than 0.95.

8.4.5 Regional Inter-Observer Variability

The left ventricle poses some segmentation difficulties regarding the mitral valve and outgoing tract regions, since there are different (valid) segmentation criteria which might include or not the outgoing tract and consider different ways of defining the segmentation close to the valve. This might result in considerable segmentation differences. For example, considering Exam 3, the lowest Williams' index values happened for Radiographer C. Different criteria used by this radiographer might explain the poorer agreement found towards the remaining radiographers. In fact, visual comparison of some of the segmentations presenting the worst agreement values confirmed that the main difference was located in the mitral valve/outgoing tract region.

Following on the segmentation differences depicted by the comparison data (i.e., Jaccard coefficient lower than 1.0) presented in figures 8.4 and 8.5, we were interested in characterizing

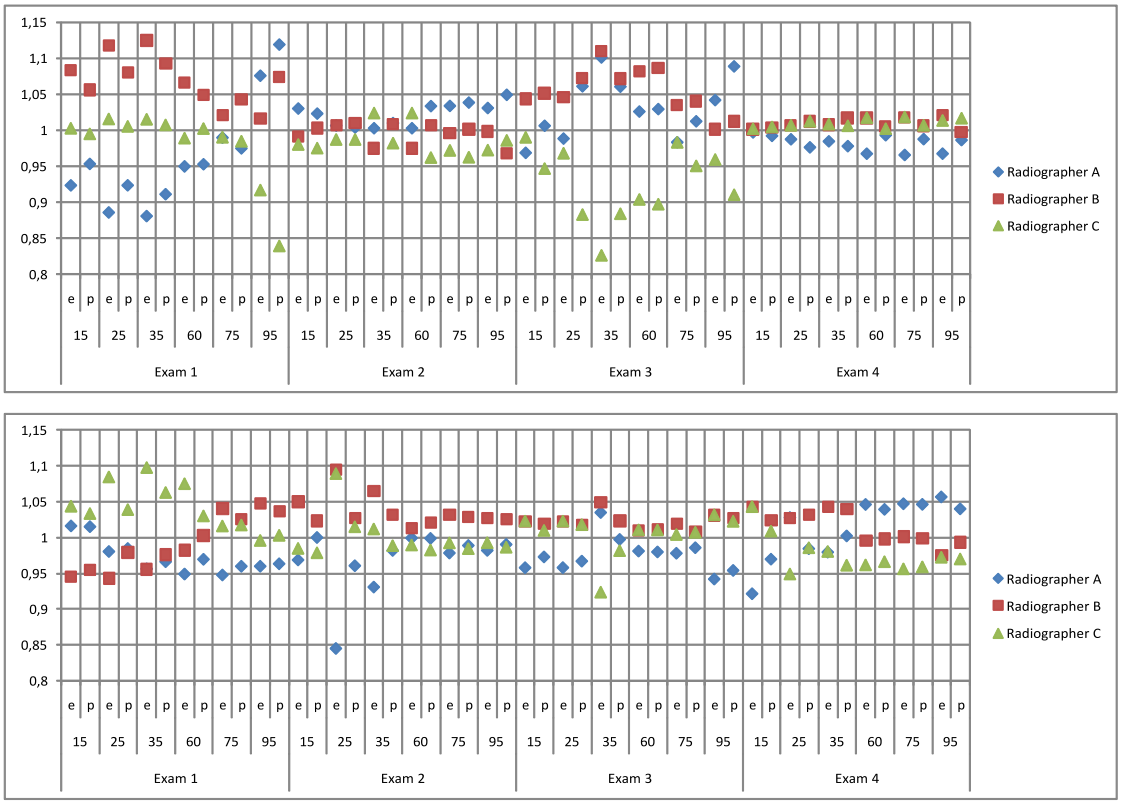


Figure 8.5 – Inter-radiographer variability: Williams' agreement index (using Jaccard coefficient as similarity measure) computed for all radiographers for each cardiac phase (e – endocardium; p – epicardium) and exam. Top, first evaluation run; bottom, second evaluation run.

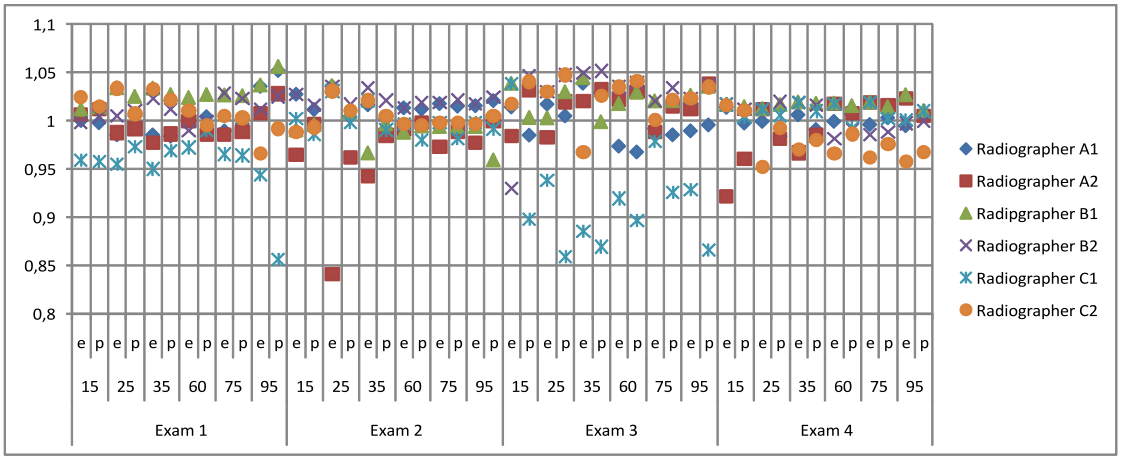


Figure 8.6 – Agreement values (using the Williams index and Jaccard coefficient as similarity measure) for each radiographer and considering both evaluation runs. For example, Radiographer A1 and Radiographer A2 are Radiographer A at the different evaluation runs.

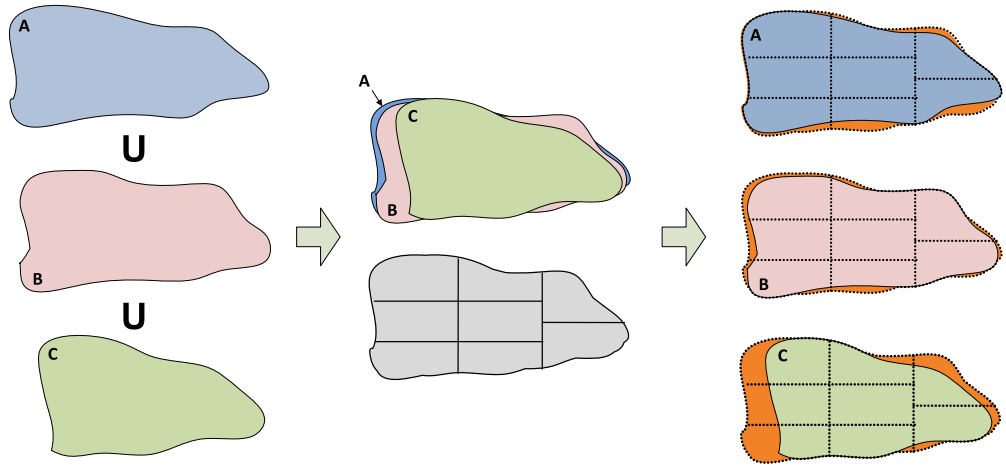


Figure 8.7 – Segmentation region analysis. Corresponding segmentations from the three radiographers are joined to form a reference. Difference percentage is computed for the individual segmentations on a regional basis. This is performed for all phases and all exams. Representation in 2D for the sake of simplicity.

differences more accurately, i.e., where the differences were located and their extent. This, we found, is not common in LV segmentation assessment. In fact, several works evaluating LV segmentations compare only total volumes between segmentations (e.g., Muhlenbruch et al. [32] and Abbara et al [58]). Yet, two segmentations might have the same volume and not correspond to the same region or they might differ in regions where that is admissible due to different acceptable segmentation criteria.

To perform this regional analysis, the usual 16 segments for myocardial analysis have been considered (excluding the apex).

For each cardiac phase (and exam) the reunion among the segmentations performed by the three radiographers was computed. Then, it was divided in 16 segments and superimposed on each segmentation. Each segment was considered individually and a comparison performed between each corresponding area in the radiographers segmentations and the reference segment. Figure 8.7 depicts this procedure. For each segment the percentage of different (missing) voxels was computed

Figure 8.8 shows the average percentage (among the three radiographers) of different voxels found for all 16 segments in each exam.

Notice that, for the first three exams, the percentage of difference towards the reference is considerably higher for the first six segments. This confirms our assumption that most error was due to differences in the basal region. For the fourth exam the difference is much smaller and the peaks on segments 10/11 and 15/16 are just around 10% and for only one of the phases.

Figure 8.9 shows the average difference percentage for each myocardial segment (and exam)

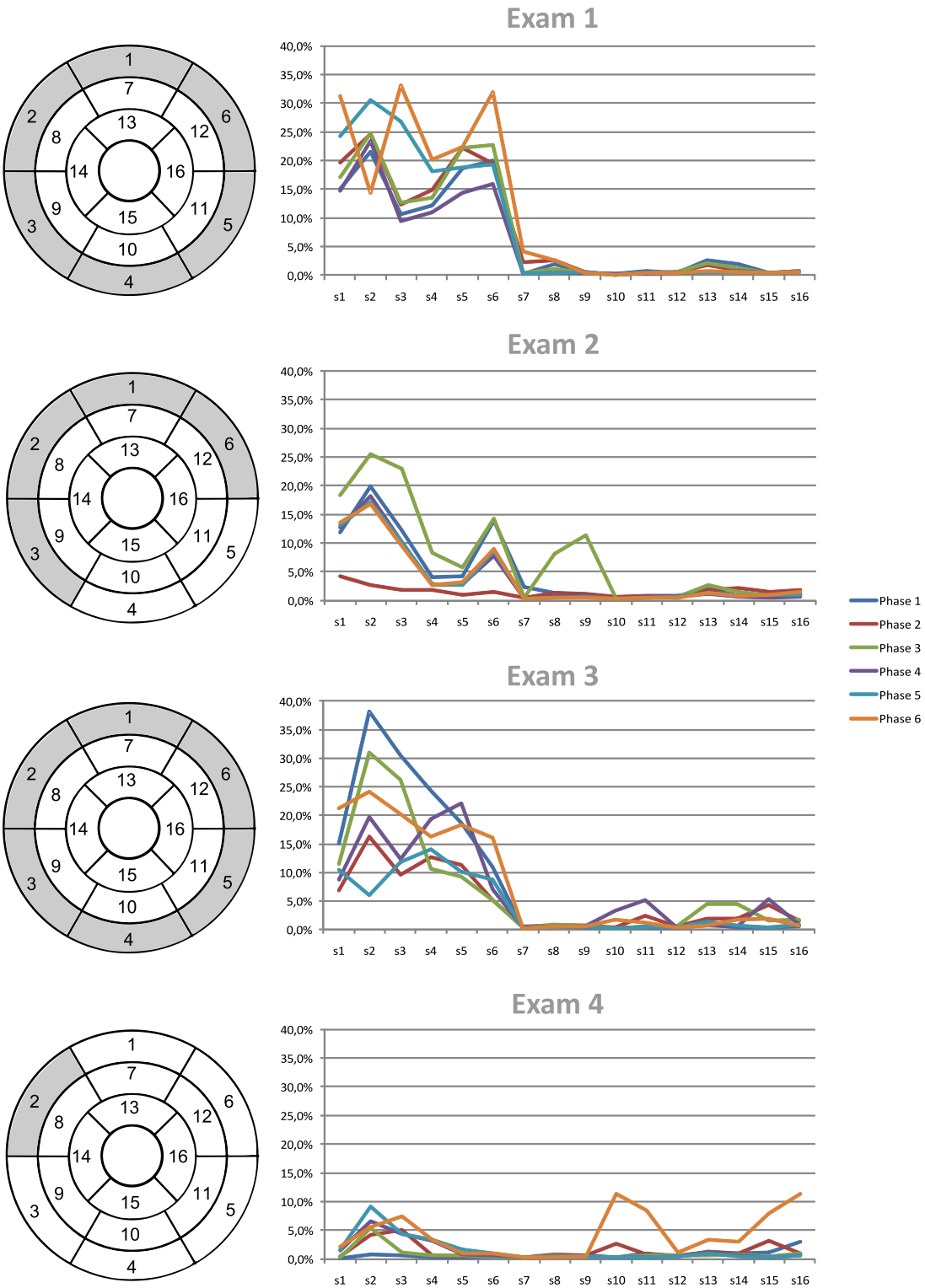


Figure 8.8 – Average regional difference for endocardium segmentations (among three radiographers) found for each segment, cardiac phase and exam. The polar maps on the left depict all segments with more than one cardiac phase presenting a difference above 5%.

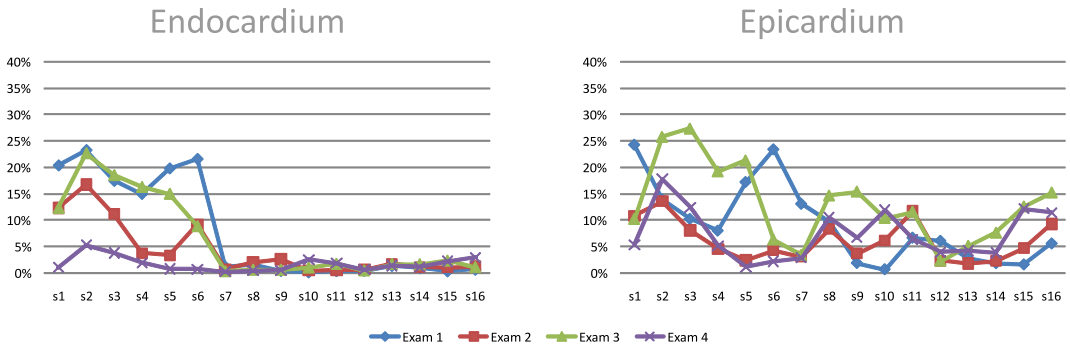


Figure 8.9 – Average regional difference for endocardium (left) and epicardium (right) segmentations, over all cardiac phases and radiographers, for each myocardial segment.

over all cardiac phases. Notice that for both endocardium and epicardium the major differences are found in the basal segments (s1 to s6) and that the epicardium shows slightly higher differences over the medial and apical segments.

8.4.6 Image quality and user satisfaction

Regarding the ratings given by the radiographers, concerning image quality and user satisfaction, table 8.4 presents the median values for each item. Since we avoided images with any acquisition artefacts a good rating concerning image quality was expected. The rating concerning epicardium segmentation shows evidence that the epicardium was considered harder to segment than the endocardium (considered easy to segment). This can be explained given that it is not often easy to identify, in the septal region, for example, where the epicardium ends and the inside of the right ventricle begins. Evidence of such difficulty has also been found, as presented in the previous sections, when comparing segmentations.

Finally, concerning the overall satisfaction the radiographers rated it positively (3.5).

Table 8.4 – Ratings given by radiographers regarding image quality and user satisfaction for both evaluation runs (e1 and e2).

Question	**Meaning**		**Median**	
	1 . . .	. . . 5	**e1**	**e2**
Exam quality	Very Poor	Very Good	4	4
Endocardium segmentation	Very Hard	Very Easy	4	4
Epicardium segmentation	Very Hard	Very Easy	3	3.5
Overall satisf. with seg.	Dissatisfied	Satisfied	3.5	3.5

8.5 Conclusions

This chapter presents the evaluation of the LV segmentation tool presented in chapter 5 (also including the editing tool presented in chapter 6).

The time taken to perform the segmentations is very reasonable when compared to recent results presented in the literature. The average segmentation times obtained using our method are similar to those reported by Muhlenbruch et al. [32] (around 170 seconds to segment two endocardia) and far lower than those reported by Coche et al. [92] who report 15-20 minutes for semi-automatic segmentation of only two cardiac phases: end-systole and end-diastole (endocardia and epicardia). On the presented evaluation, radiographers took around 13 minutes to segment 6 cardiac phases (6 endocardia + 6 epicardia).

Regarding the segmentations, a summary of the average and median values obtained using the Jaccard coefficient and the William's agreement index for the different evaluated aspects is presented in table 8.5. When applicable, the values presented are the worst obtained (generally from the first evaluation run). The overall values have been computed considering the segmentations for both evaluation runs.

Table 8.5 – Summary of mean and median of comparison/agreement results obtained during the evaluation. When applicable the values presented are the worst obtained (generally, the first evaluation run). Overall values refer to a comparison considering segmentations for both evaluation runs.

Endocardium

	Jaccard		**Williams Index**	
	mean	med	mean	med
editing	0.97	0.99	–	–
intra-obs.	0.95	0.96	–	–
inter-obs.	0.93	0.93	1.00	1.00
overall	0.93	0.94	1.00	1.01

Epicardium

	Jaccard		**Williams Index**	
	mean	med	mean	med
editing	0.98	0.99	–	–
intra-obs.	0.93	0.94	–	–
inter-obs.	0.91	0.91	1.00	1.00
overall	0.92	0.93	1.00	1.00

The editing performed by the radiographers, over the first segmentations provided by the segmentation tool, amounts to a very small percentage of the initial volume, for most cases. The exception is the reference phase, as expected (refer to figure 8.2), since the corrections performed on this phase will frequently improve the segmentations for the remaining cardiac phases. This might be an indication that radiographers might not need to supervise all cardiac phases, which could be used to speed the segmentation process by just presenting radiographers the reference phase. Nevertheless, further validation is needed.

Intra-radiographer variability is not very pronounced with the Jaccard coefficient consistently above 0.90 for two of the radiographers and above 0.85 for the full radiographers set. The average value is above 0.90 for both endocardium and epicardium (table 8.5).

Inter-radiographer variability is also low, with the Jaccard coefficient above 0.85 (figure 8.4) and quite close to the comparison values obtained for the intra-radiographer variability. This is also noticeable considering the Williams index (figure 8.5) and the average values presented in table 8.5. Further analysis has shown that, specially for the endocardium, most differences are placed at the basal region. Concerning these differences, due to the definition of the mitral valve plane and inclusion of the outgoing tract, one possible approach might be to impose a more strict segmentation criteria, e.g., asking the user to set the mitral valve plane and use that same plane to limit the segmentation towards the outgoing tract without allowing editing. Another approach might be to define a segmentation criteria among all radiographers in order to keep better coherence between segmentations. A conjugation of these two approaches, while allowing some editing, might be the best choice.

Concerning user satisfaction there is still room for improvement. Talking with the radiographers, one possible reason for this moderate level of satisfaction might be their unfamiliarity with the 3D editing tool (chapter 6) since no similar tool exists in the workstations they use daily. Satisfaction might be improved by additional training, resulting in greater user proficiency and confidence or, if required, tool improvement. Satisfaction levels between the first and second evaluation runs have not improved. This was expected since this problem was detected after the first evaluation run and no specific training was performed in between in order not to influence the second evaluation run.

Finally, a more elaborate statistical analysis should be performed to further explore the obtained data and the test data sets expanded to encompass exams from more patients.

Part III

Left Ventricle Analysis

Chapter 9

Left-Ventricle Global and Regional Functional Analysis

> "Há muito que [Bergson] mostrou o perigo do desenvolvimento anormal da maquinaria se o espírito não vier animar esse complemento do organismo humano."
>
> – Diamantino Martins

EXPLORING left-ventricle functional analysis is of paramount importance for the assessment of coronary artery disease. Even though coronary CT angiography provides data concerning multiple cardiac phases, along the cardiac cycle, LV function is usually evaluated using just the end-systolic and end-diastolic phases. This unused wealth of data, mostly due to its complexity and the lack of proper tools, has still to be explored in order to assess if further insight is possible regarding regional LV functional analysis. Furthermore, different parameters can be computed to characterize LV function and while some are well know by clinicians others still need to be tested concerning their value in clinical scenarios.

Based on these premises we present several parameters characterizing global and regional left ventricle function, computed for several cardiac phases over one cardiac cycle. The data provided by the computed parameters is shown using a set of visualizations allowing synchronized visual exploration of the different data. The main purpose is to provide means for clinicians to explore the data and gather insight over their meaning and their correlation with each other and with diagnosis outcomes.

The work presented in this chapter has been partially published in:

- Samuel Silva, Beatriz Sousa Santos, Joaquim Madeira, "Exploring Different Parameters to Assess Left Ventricle Global and Regional Functional Analysis from Coronary CT Angiography", Computer Graphics Forum, accepted for publication, 2012
- Samuel Silva, Beatriz Sousa Santos, Joaquim Madeira, "Left-Ventricle Global and Regional Functional Analysis from MDCT Images", Proc. Ibero-American Symposium on Computer Graphics (SIACG11), pp. 217–223, Faro, Portugal, June, 2011 (**2**nd **Best Paper Award**)

9.1 Introduction

Left-ventricle functional analysis is of paramount importance for the assessment of coronary artery disease. Non-invasive coronary artery imaging is currently possible using MDCT scanners [96]. For this purpose the patient is injected with a contrast agent which improves the contrast of the heart chambers (atria and ventricles) and coronary arteries. Typically, the data provided by these exams consist in a set of 10 image volumes evenly distributed along the cardiac cycle. Each of these image volumes encompasses the full heart and therefore includes data concerning not only the coronary arteries but both atria and ventricles.

Scanner technology has been advancing at a fast pace. Several studies have shown [2, 53, 52] that 64–MDCT imaging is clinically acceptable as a source to evaluate LV function when compared to other image modalities such as cardiac magnetic resonance (CMR), echocardiography and single-positron emission computed tomography (SPECT) and is even used as a reference in some studies [54].

Apart from a few works concerning CMR images, which explore several parameters (e.g., wall thickening and regional ejection fraction) and analysis methods (e.g., plots for regional mean values) over time, LV function analysis has been limited to computing global/regional data for the systolic and diastolic phases not taking advantage of the cardiac data available in-between. This holds true with CTA images, in spite of the multiple cardiac phases available, due to the complexity of the data and the lack of adequate tools [97].

Therefore, the question remains if further insight into LV function is possible if this additional data is considered. To answer this question three important stages should be considered: 1) LV data must be segmented for all cardiac phases; 2) different parameters deemed important to characterize global and regional LV function must be computed and presented and 3) these tools should be tested in clinical scenarios to assess their applicability.

Several parameters can be computed for the different cardiac phases. While some have a very clear meaning to clinicians, other have yet to be explored in order to assess their significance and, more importantly, their correlation with each other and with different cardiac pathologies. Therefore, a framework should be provided to clinicians in order to allow coordinated views for analysis, using different parameters and visualization options. This framework should encourage exploring the data thus provided and, profiting on the clinicians knowledge regarding physiology and anatomy, help devise how it can be used to support diagnosis. Needless to say, this is also an important step towards using automatically computed parameters to provide automated diagnosis[98].

The work presented in this chapter concerns global and regional functional analysis of the LV from 4D MDCT cardiac images based on how different parameters characterizing LV function vary

over time. The described features have been implemented in CardioAnalyzer [1], a tool developed using Qt[2], MITK[3], ITK[4] and OpenGl [5].

The main contributions of the work presented in this chapter are:

- Left ventricle analysis from CTA images based on multiple cardiac phases
- Framework presenting different parameters characterizing left ventricle function
- Simultaneous visualization of multiple user chosen parameters
- Link between the different visualizations and cardiac images (brush + link [99]) to support visual exploration of the provided data

This chapter is structured as follows. Section 2 provides a short overview on related work concerning LV functional analysis. Section 3 presents the visualizations provided and how they relate with the LV and its anatomy. Section 4 briefly describes the LV segmentation method, used to obtain LV data for analysis. In section 5 follows a description of the different parameters computed to characterize global and regional LV function along with some discussion on how they (and corresponding visualizations) might provide hints regarding abnormal LV function. Finally, in section 6, we present a discussion of the work carried out and conclusions.

9.2 Related Work

Regarding functional analysis of the LV from CTA images, the works presented in the literature tend to use just two cardiac phases, the end-systolic and end-diastolic phases, neglecting the remaining phases. Their main purpose is usually the (successful) validation of CTA based analysis towards other image modalities (e.g., 2D echocardiography) and no attempt is performed to go beyond the common parameters such as blood volumes for the end-systolic and end-diastolic phases and ejection fraction. A few recent publications can serve as an example. Fischbach et al. [51] assessed global and regional LV function based solely on the end-systolic and end-diastolic phases. Adding to the computation of the ejection fraction they characterized wall motion/thickness by performing analysis of short-axis slices from both cardiac phases. Similar work has been presented by Palazzuoli et al. [100].

[1] Refer to appendix B for additional details.
[2] Qt library: `http://qt.nokia.com/`
[3] Medical Imaging Interaction Toolkit: `http://www.mitk.org`
[4] Insight Segmentation and Registration Toolkit: `http://www.itk.org`
[5] OpenGL library: `http://www.opengl.org`

Sarwar et al. [53] perform LV functional analysis also based on the end-systolic and end-diastolic phases. In their study, apart the typical blood volumes and derived ejection fraction they perform a qualitative assessment of regional LV function based on wall motion.

Ionasec et al. [97] have performed analysis of cardiac data from CTA images (and transthoracic echocardiography) concerning the aortic and mitral valves. Although our work does not concern cardiac valves it is still important to notice that Ionasec and colleagues performed an analysis over time presenting line graphs of different parameters (e.g., valve area) along the different cardiac phases.

From the examples provided above it is clear that CTA based analysis of LV function is feasible and gathers interest from the clinical community but, apart from the last example, which is not closely related with the subject matter of our work (LV analysis), no work seems to explore the full dataset provided by CTA.

On the other hand, when considering LV analysis using CMR, a few works have been presented which allow exploring cardiac data over the full cardiac cycle.

Wesarg [101] presents LV analysis based on different parameters over time. The author uses polar maps to depict global results for each of the parameters which have a corresponding line graph showing how these parameters vary over time for each myocardial segment. According to Wesarg, both representations showed evidence of existing dysfunction (when it existed), e.g., by a flattening of the curves in the line graphs.

Kermani et al. [102] present a method for quantitative analysis of LV performance based on multiple cardiac phases. The method allows the assessment of myocardium motion and various representations are provided concerning strain percentage for each myocardial segment over time. Polar maps are used to show overall variation of wall motion. A very interesting aspect of this work, apart from the proposed quantitative method, is how the authors assess normal motion patterns from healthy patients and then use them to normalize the results.

Recently, Wesarg [103] has proposed a method to assess ventricular reduction surgery outcomes which is based on the presentation of a series of polar maps showing endocardial distances and regional blood volumes over time. This is, apart from the work by Breeuer [104] (also for CMR images), the only work presented in the literature paying attention to analysis over time by providing a sequence of polar maps, one of the features we provide in the presented work but for CTA.

To summarize, to the best of our knowledge, no work has been presented in the literature which provides LV functional analysis from CTA exploring the data available between systole and diastole. Furthermore, even considering CMR, no work supports the simultaneous exploration of different parameters characterizing LV function. This, we argue, should be performed providing

coordination between the different visualizations and the cardiac image.

9.3 Visualizations

First, to understand the analysis and representations presented along this chapter it is important to consider how the LV is divided in different segments and how the data concerning each segment is usually represented (as described in chapter 2). In what follows, a few details are provided concerning the visualizations used to depict the computed data along with a discussion concerning the reasons that support our choices.

9.3.1 Representations and Interaction

This section provides a description of the representations chosen for the computed data along with the different interaction options provided.

Polar Maps

The polar map (bull's eye), presented in chapter 2, is a standard, well known way of representing LV analysis data [102]. In a simple representation it is possible to have a global view of what is happening in the different regions. It is also possible to easily connect each segment to its location in the LV structure. Therefore, we adopted the polar map as the main representation for the data provided by the different parameters.

Representing regional data means colouring each segment in the polar map according to a single scalar (e.g., figure 9.11). On the other hand, representing local data requires that multiple values are presented along the polar map. The local values computed for each LV slice correspond to a circle in the polar map. Since the polar map representation is "continuous" and we are representing sampled data (further details ahead), colour interpolation is used between data points. Figure 9.1 presents the different steps involved in building a polar map representation for local data.

No abrupt changes are expected in LV shape and properties. Nevertheless, due to image noise or acquisition artefacts, the computed local parameters present some noise which often results in polar maps with numerous spurious changes in colour. When presented with such polar maps clinicians tend to find them awkward. To tackle such issue, polar maps can be smoothed by applying a Gaussian filter to the data.

Figure 9.2 shows two polar maps built for the same data set. Gaussian filtering has been applied to the data before building the polar map on the right, resulting in a smoother transition between the various regions. The amount of admissible smoothing (i.e., without distorting relevant

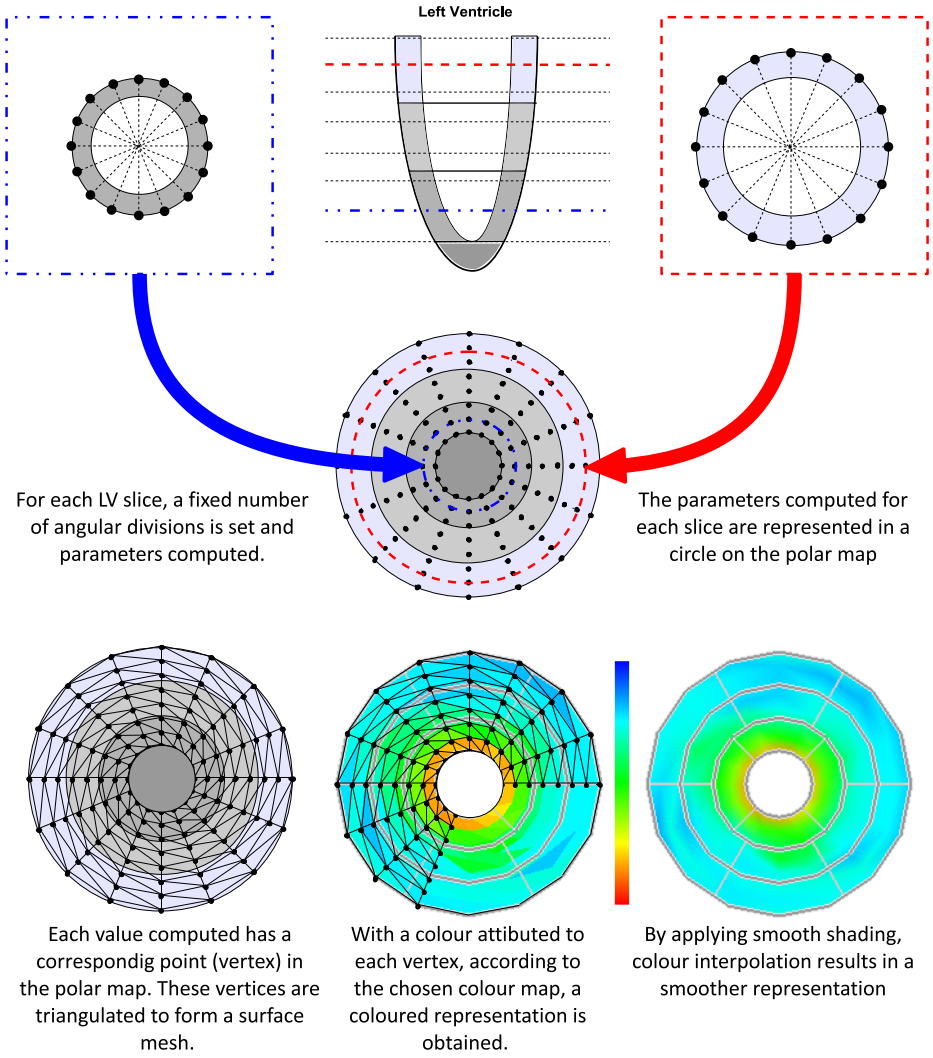


Figure 9.1 – Different steps involved in building a polar map representation for local data. Analysis is performed for several slices along the LV long-axis and, for each slice, at chosen angular steps. The presented example has been computed for eight slices and 16 points per slice.

features of the data) was set by observing different polar maps with different smoothing levels and choosing the one providing the smoother, less distorted (i.e., keeping important features), representation of the original.

Line Graphs

Line graphs are also used to present global and regional data for the different cardiac phases. For example, figure 9.3 shows, on the rightmost line graph, blood volumes for each cardiac phase, while the remaining line graphs show one curve for each myocardial segment along the different

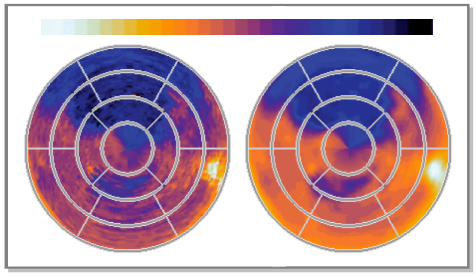


Figure 9.2 – Filtering data before representing it results in smoother polar maps removing spurious changes due to image noise or acquisition artefacts. The polar map on the right is a smoothed version of the polar map presented on the left.

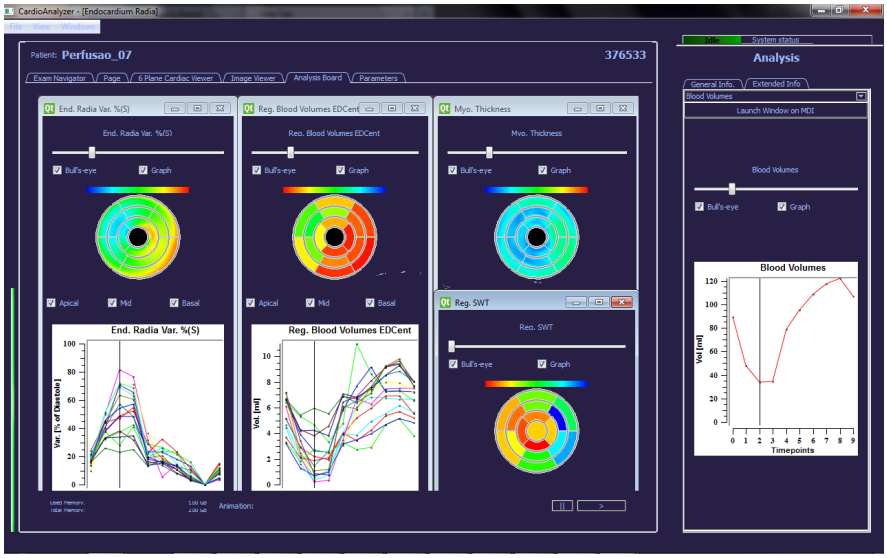


Figure 9.3 – Screenshot of CardioAnalyzer showing several analysis windows. All representations are synchronized.

cardiac phases. In order to allow easier analysis/comparison of the curves part of the same annular tomographic region (basal, mid-ventricular, apical) the user can choose which region(s) to see by checking/unchecking the corresponding checkboxes.

3D Model of the Left Ventricle

Another possible visualization could be supported by a 3D model of the LV (figure 9.4), over which analysis data could be represented. However no immediate advantage of using a 3D representation could be devised considering the main goals of our work. First of all, a visualization supported by a 3D model of the LV naturally suffers from occlusion, i.e., when inspecting from one viewpoint, the remaining data is hidden [105]. Even considering that by representing data over the 3D model we are making a clear association between data and its location (which is already provided

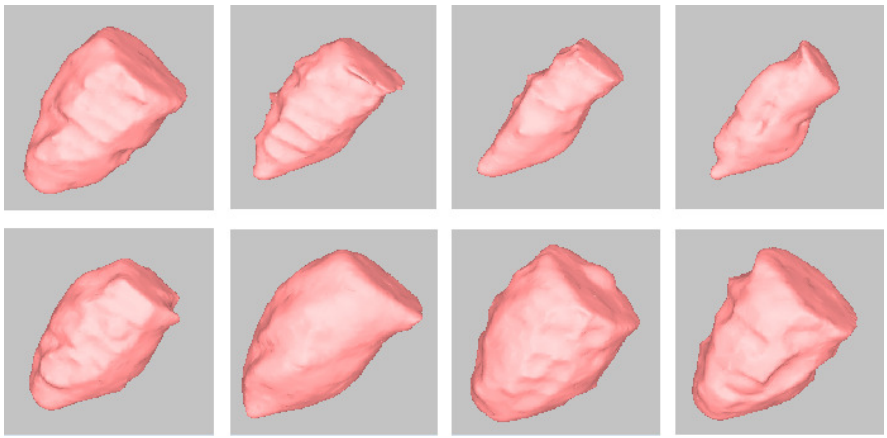


Figure 9.4 – Different 3D models created for the LV blood pool along the cardiac cycle.

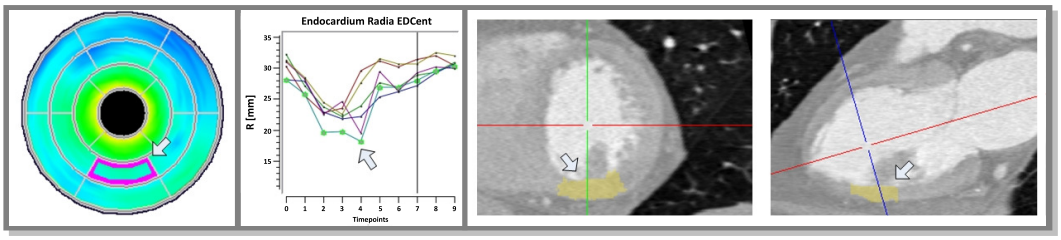


Figure 9.5 – Selecting a segment on a polar map representation: all polar maps will show the selection, all line graphs will have the corresponding curve highlighted and the anatomical region is depicted over the LV image.

by the polar map), the fact that the LV is symmetric around its long axis requires providing additional context: for example, presenting the 3D over the image or representing additional structures in the 3D to help match the 3D to the anatomy (e.g., semi-transparent right ventricle). Furthermore, evidence seems to show that 2D visualizations have, in general, an advantage over 3D visualizations for fostering knowledge [106, 107]. Nevertheless, we recognize that a 3D view of the LV might provide a quick (albeit qualitative and viewpoint dependant) assessment of shape. Therefore we provide it, without any data represented over it (in order not to mask shape), for inspection.

Simultaneous Visual Exploration

Since we aim to provide users with different parameters characterizing LV function and an environment where they can gather additional insight over each parameter and the correlation among them, one feature we consider extremely important is simultaneous visual exploration [98] of different parameters. Visual working memory has been shown to have a very short capacity con-

cerning the amount of data it can store and the amount of time that data lasts [108]. In result, performance will be hindered if users need to compare visualizations by alternating between them, even if spatial coherence is kept. Therefore, we provide an analysis board (figure 9.3) where the user can manage individual windows showing data for different parameters. The user is free to choose which windows are visible, how they are arranged and if both polar map and line graph are visible or just one of them.

Synchronized Viewing

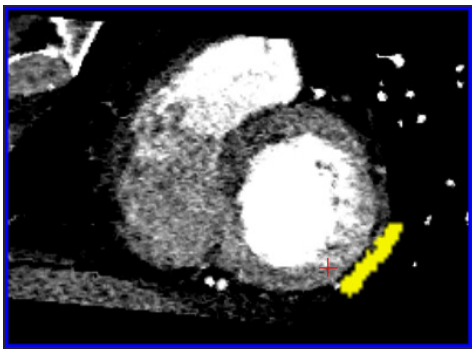


Figure 9.6 – Segment highlight shown on the exterior of the myocardium. This has the advantage of not occluding the myocardium during analysis that needs proper myocardium attenuation visualization, e.g., myocardial perfusion.

Since different parameters and visualizations can be analysed at the same time additional features are required to allow synchronized viewing of the different polar maps and line graphs and provide a link between the different representations and the anatomical region, in the images, to which a particular subset of the data corresponds (often referred to as brushing and linking [99]).

When a cardiac phase is selected, the corresponding polar maps for all selected parameters are shown and a vertical line is presented in each line graph as seen in figure 9.3.

When a segment in the polar map is selected, it is automatically selected in all polar maps and the corresponding curve, in each line graph, is also highlighted. Furthermore, the anatomical region associated with the segment is highlighted over the image data. Figure 9.5 shows a polar map with segment 10 selected, the corresponding line graph with the highlighted curve and a short-axis and two-chambers view of the LV with the highlighted region.

An alternative for highlighting the anatomical region is to show a coloured border on the corresponding segment instead of an over imposed mask (see figure 9.6). This might be particularly important if visualizing the myocardium can offer additional data to the clinician, e.g., when assessing myocardial perfusion.

Colour Scales

Regarding the colour scales used, the rainbow scale, presented in some of the polar maps, is known to pose several problems in visualizations [109]. The reasons why we kept using it in this

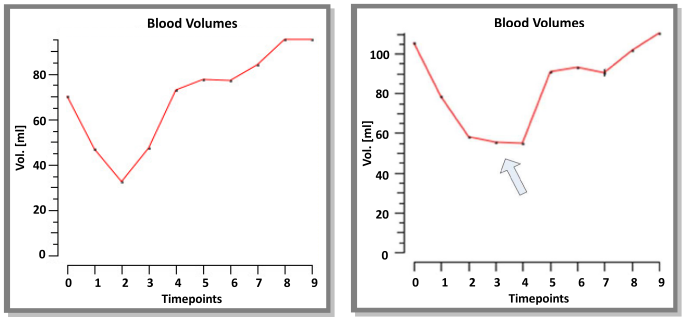


Figure 9.7 – Left ventricle global analysis showing blood volumes for each cardiac phase for two patients. Notice that the line graph on the right exhibits a rather difficult to identify end-systolic phase.

work, are mainly two. The first concerns clinician familiarity with this colour scale and therefore it is easier for them to interpret the polar maps. The second reason has to do with the analysis clinicians often perform: in fact, they are mostly concerned with a qualitative evaluation of the polar map and extreme values (in blue and red) are the most important. The values in-between are also important but no proper quantification is required (which would be difficult with the rainbow colour scale given that it is not perceptually uniform). The typical colour scale can also vary depending on the application. For example, for myocardial perfusion using single-positron emitting CT (SPECT), a different colour scale is typically used and should be kept, in similar contexts (e.g., myocardial perfusion CT), to profit from user familiarity.

Cine-Analysis

Visualizing the images and polar maps for each cardiac phase or each of the cardiac volumes along the cardiac cycle can sometimes be improved if they are presented in an animated (self-repeating) sequence. This allows the user to detect abnormal situations (e.g., in heart movement), and enhances the way variations can be perceived analysing the polar maps. During the animated sequence (speed adjustable by the user) all representations (3D models included) keep synchronized with the current image volume. The line graphs present a vertical marker to depict the current cardiac phase.

9.4 Left Ventricle Analysis

The following sections describe several parameters provided for global, regional and local LV analysis. These parameters provide data on how the LV is pumping blood, how it is contracting and how blood is reaching the myocardium (perfusion).

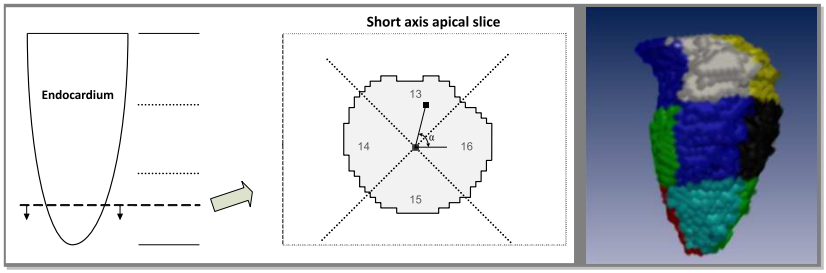


Figure 9.8 – Assigning each voxel in the endocardium voxel mask to a LV segment: each short-axis slice is processed; the slice number and the angle computed for each voxel determines its segment. Right, the endocardium voxel mask shown with segments in different colors (partial view).

9.4.1 Left Ventricle Global Analysis

The blood volumes for each cardiac phase are computed by counting the number of voxels in each segmented volume and multiplying by voxel volume according to image spacing. Having the blood volumes for all cardiac phases, it is possible to determine the end-systolic (lowest blood volume) and end-diastolic (highest blood volume) phases. Figure 9.7 shows line graphs depicting the blood volume for each cardiac phase (timepoints) for two patients. While in the first line graph the blood volume reaches a clear low at timepoint 2 (thus the end-systolic phase) in the bottom line graph the end-systolic phase is not as clearly apart from the neighbour phases.

The ejection fraction (EF) can then be computed as described in chapter 2.

9.4.2 Left Ventricle Local and Regional Analysis

For LV regional analysis data concerning multiple LV regions is computed. Two different approaches are possible for regional analysis: one includes computing local data for the LV and another consists in computing data for each LV segment. In those situations where local measures are possible, both approaches can be used since an average value for each segment can be computed from the different local values available. For example, endocardium radius can be computed locally and a mean radius can be computed for each LV segment. On the other hand, some measures, e.g., blood volume, only make sense as a regional parameter. Furthermore, there are commonly used parameters, such as the systolic wall thickening (SWT) [110] which require the computation of local and regional values.

Regional Blood Volumes and Regional Ejection Fraction

Regional blood volumes are obtained by counting the number of voxels in each segment and then multiplying by voxel volume.

To assign the voxels to each segment, analysis is performed on short-axis slices. For each slice, the centroid of the voxel mask is computed and, for each voxel, the angle at which it is positioned relative to the horizontal (see Figure 9.8) is determined. The slice number is used to assign the voxel to a annular tomographic region (basal, mid-ventricular or apical) and set the correct number of segments expected in that region. Since segment numbering does not start at angle zero, an angle offset is also needed. A better alternative to the currently analysed voxel mask centroid is the corresponding centroid for the diastolic phase. This way, blood volumes are computed defining LV segments according to the LV long axis at rest.

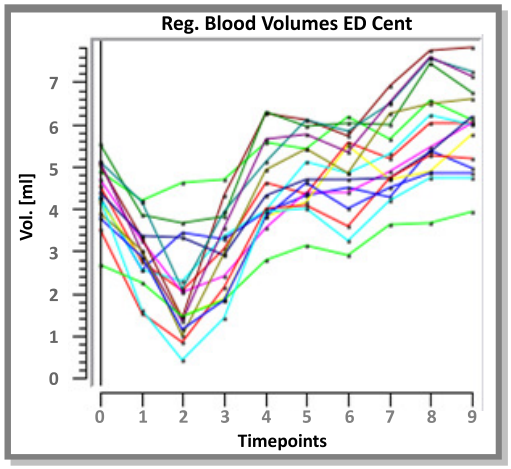


Figure 9.9 – Regional blood volumes graph. Each of the curves represents blood volumes for each myocardial segment along the cardiac cycle.

By computing regional blood volumes for all available cardiac phases, it is possible to obtain an overview of their variation along the cardiac cycle. Figure 9.9 shows a line graph depicting several curves for the amount of blood found in each segment along the cardiac cycle.

A representation is also possible using a polar map for each cardiac phase depicting the regional blood volumes as shown in figure 9.10.

Regional ejection fraction is computed considering the regional blood volumes for the end-systolic phase and the corresponding regional blood volumes determined for the diastolic phase. The polar maps in figure 9.11 depict the regional ejection fraction for an healthy patient (left) and for a patient with diagnosed low myocardium contractility (right). Notice the very clear prevalence of red in the second polar map, evidence of very low regional ejection fractions.

Endocardium Radius

Given an endocardium segmentation it is possible to compute its radius and, by analysing how it changes along the cardiac cycle, obtain data regarding myocardium wall motion. This is based on the fact that maximum myocardium displacement during systole occurs in the endocardium [111].

For each short-axis slice the endocardium centroid is obtained and the radius is computed at constant angular intervals (see figure 9.12). A trade-off between the number of angular steps (and considered slices) and computation speed is performed considering that the endocardium is not prone to present abrupt changes. The data presented in this chapter has been computed using 25

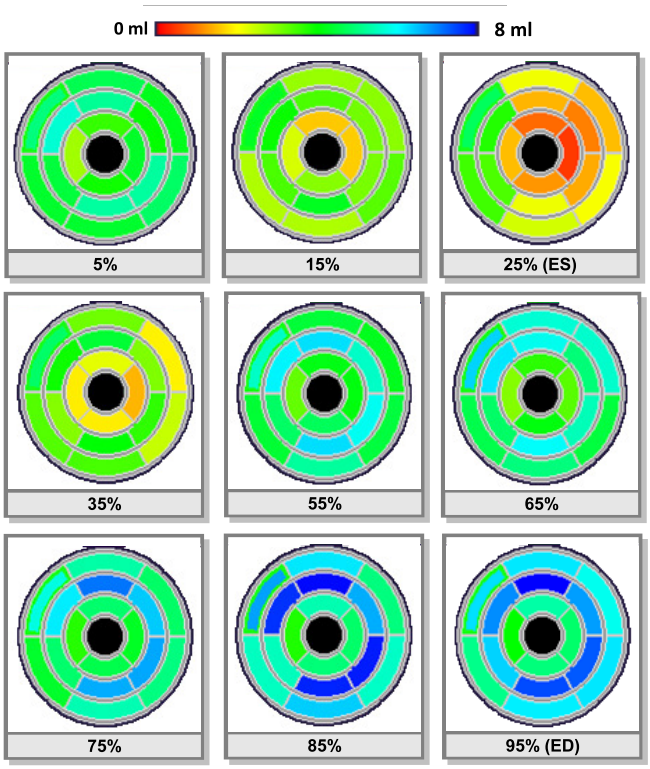


Figure 9.10 – Regional blood volumes for each available cardiac phase. End-systolic (ES) and end-diastolic (ED) phases automatically defined according to total blood volumes.

angular steps per slice and analysis was performed for a slice in every four slices (around 25 slices). The image is travelled along each angular direction looking for the border of the endocardium voxel mask. Then, the radius is converted from image units to world units considering image spacing

$$R(mm) = \sqrt{((v_x - c_x) \times s_x)^2 + ((v_y - c_y) \times s_y)^2}$$

where (c_x, c_y) are the centroid coordinates, (v_x, v_y) are the border voxel coordinates and s_x and s_y concern image spacing as depicted in figure 9.12. Even though the segmentation method includes a hole filling step and all the segmentations are supervised there is always the chance of a "hole" in the mask. To avoid a badly computed radius, when what is thought to be a border voxel is reached the image keeps being travelled in the same direction for some additional voxels. If an active voxel is found the algorithm goes back to looking for a border voxel. Otherwise, the border voxel initially found is considered.

Using the endocardium centroids computed for each short-axis slice might not give a proper idea concerning radius variation asymmetry. This is due to the fact that the reference point (centroid) is always computed from the currently analysed slice. Therefore, it does not provide a

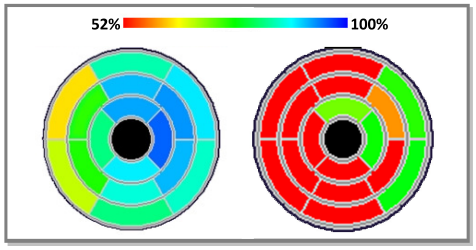


Figure 9.11 – Regional ejection fraction. Left, polar map for healthy patient. Right, polar map for patient with depressed ejection fraction.

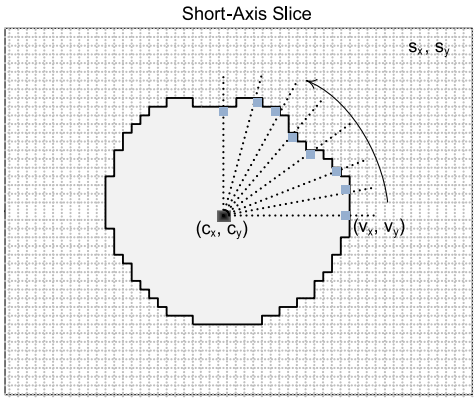


Figure 9.12 – Endocardium radius computation. Each short-axis image is traveled along different directions looking for border voxels.(c_x, c_y): centroid coordinates; (v_x, v_y): border voxel coordinates; s_x, s_y: image spacing.

global reference to be used for all phases. In order to explore radius asymmetries, the centroids for the diastolic phase (when the heart is at rest) are used to compute endocardium radii for all cardiac phases, thus considering the LV long-axis at rest.

Figure 9.13 shows the polar maps for endocardium radius using, on top, each phase centroids and, on the bottom, diastolic phase centroids. Notice how on the bottom polar maps it is possible to see some radius asymmetry (smaller radius on the right side of the polar map) in the end-systolic (ES) phase.

Figure 9.14 shows the mean segmental endocardium radius values over the different cardiac phases. In order to allow an easier analysis the user can choose to view only the curves associated with each annular tomographic region (apical, mid-cavity or basal segments) or any combination of the three.

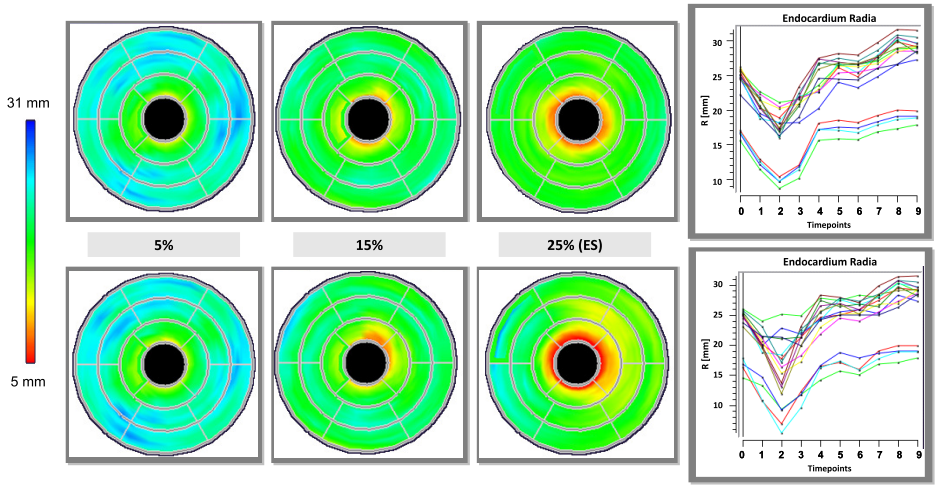


Figure 9.13 – Polar maps depicting endocardium radius for different cardiac phases. Top, endocardium radius has been computed using the centroids for each cardiac phase; bottom: endocardium radius has been computed using the diastolic phase centroids for all phases. Notice how the polar maps and the line graphs on the right present noticeable differences, particularly around the end-systolic phase (timepoint 2).

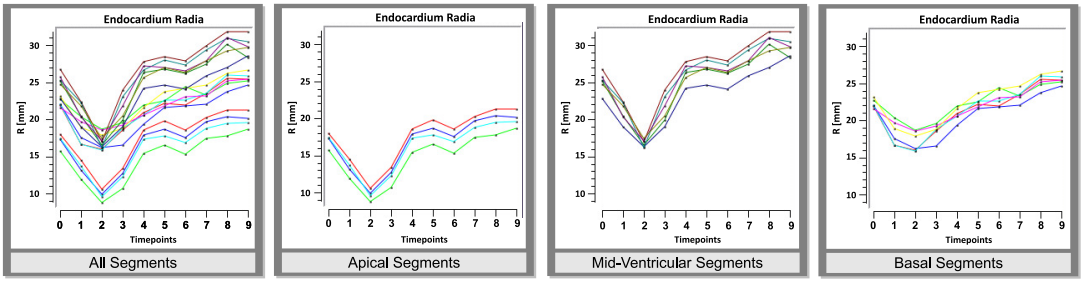


Figure 9.14 – Line graphs showing the evolution of segmental mean endocardium radius along the cardiac cycle. Apical, mid-ventricular and basal segmental curves can be viewed separately.

Endocardium Radius Variation

Given the shape of the endocardium, the radius is larger in the basal region and smaller in the apical region. Therefore, when representing the computed radius in a polar map using, for example, a rainbow colour scale, the apical region will always limit the perception of how the radius varies on the remaining regions. One alternative would be to represent radius variation (regarding the corresponding radius in the diastolic phase) but this would result in a similar problem with the basal region, where radius variation is higher, hindering the analysis of the apical region. Thus, radius variation is expressed as a percentage of the corresponding radius in the diastolic phase which results in a normalization of the variation for all regions.

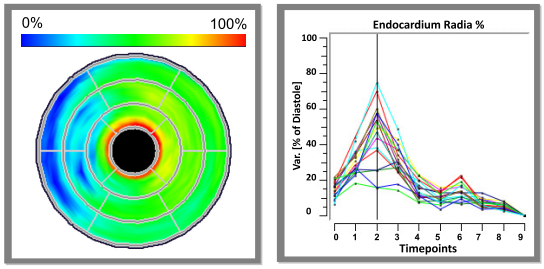


Figure 9.15 – Polar map depicting endocardium radius variation for the end-systolic phase and segmental mean radius variation curves.

Comparing the end-systolic polar maps in figures 9.13 and 9.15, drawn for the same exam data, it is clear that representing the radius variation better highlights segments (on the left of the polar map) with smaller variation while leaving the remaining segments with an homogeneous colour.

Myocardium Wall Thickness

To compute myocardium thickness, myocardium voxel mask is obtained by subtracting the endocardium voxel mask from the epicardium voxel mask. Myocardium thickness is computed using a similar method as the one described for endocardium radius computation. Each short-axis slice is analysed and, using the corresponding end-diastolic phase centroid, myocardium thickness is computed at constant angular steps.

Having myocardium wall thicknesses, systolic wall thickening (SWT) [110] can be computed as

$$SWT = 100 \times \frac{WT_{ES} - WT_{ED}}{WT_{ES}}$$

where WT_{ES} and WT_{ED} are the wall thickness for the end-systolic and end-diastolic cardiac phases.

It is important to note that the method used to compute wall thickness provides good results for the basal and mid-ventricular regions but might result in a slight thickness overestimation in the apical region [112] and methods have been presented in the literature which might help improve our approach (e.g., Senesi et al. [113]).

9.5 Discussion and Conclusions

This chapter presents different parameters characterizing global and regional LV function computed from 4D MDCT images of the heart. The computed data is represented using line graphs

presenting mean segmental values along time and synchronized polar maps depicting a coloured representation for each cardiac phase.

The different parameters allow a characterization of various aspects of LV cardiac function and several interesting outcomes can be highlighted. Notice, for example, how the curves for endocardium radius or regional blood volumes over time follow a common pattern (with different offset between annular tomographic regions as seen in figure 9.14). If one of the segmental curves stands out it might be due to a segmental problem (see line graph in figure 9.5). Notice also how in figure 9.7 the curve on the line graph on the right shows a different behaviour around the end-systolic phase which would not be clear if only a global parameter such as the ejection fraction was used. Furthermore, providing a "connection" to the anatomical region associated with the different segments allows quickly checking the anatomy to, for example, rule out abnormal parameters due to a segmentation problem.

Left ventricle 3D models (see figure 9.4), already available to the user, might also be used as an alternative support to represent the computed parameters but it is important to notice, as stated before, that a 3D model does not offer a "whole ventricle view" as the one provided by the polar map and, besides, the latter is well known by clinicians. Furthermore, it should not be forgotten that, given LV symmetry around its long axis, presenting the 3D models requires that a context is provided (e.g., 3D model and image) to have a correspondence with anatomical regions.

Informal evaluation of the presented features has been carried out with the help of a clinician with daily experience in cardiac image analysis. He regards the provided features, in particular myocardium thickness, potentially important for diagnosis, since they provide indication of possible infarction regions. Myocardium thickness might also be advantageous when combined with myocardial perfusion assessment data[6].

Additional parameters might also be tested for LV functional analysis. As highlighted earlier, Kernani et al. [102], present a method to correct the display of motion patterns by considering those patterns for a set of healthy patients and then normalizing the display for the remaining patients. This allows to deal with cases when, for example, a parameter (e.g., myocardium radius variation) is naturally lower/higher in a particular region than the mean value for the whole ventricle. If this is not corrected then that region might be depicted as having a problem. Clinicians are aware of such situations (e.g., in the septal region, between ventricles) but correcting them might help improve the representations.

Considering the 3D models for the endocardium/epicardium it is possible to compute surface curvature which seems to help find evidence of ischemic dilated cardiomyopathy [114].

[6] Chapter 10 provides additional detail on myocardial perfusion assessment

Other visualization alternatives might also be used. The selected myocardial segment can be depicted, as shown, using a coloured region outside of the epicardium. A similar method can be used to present regional data superimposed on the image using coloured areas outside the LV. This could also be extended to show local data although care should be taken since, to reduce computational cost, local data is not computed for all slices. The option of superimposing data over the myocardium itself can also be interesting in some scenarios (refer to Silva et al. [115] for an example concerning myocardial perfusion assessment).

Our next goal is to perform clinical validation to assess how the different polar maps and line graphs can help during analysis and diagnosis. This requires that knowledge concerning LV physiology and anatomy is used to interpret the different polar maps and line graphs.

Testing the different parameters in a clinical scenario might also profit if both automatic parameters and clinician assessment can be systematically gathered in a database. For example, the clinician should be able to assign a pathology to the exam or anatomical region (e.g., myocardial segment). This would allow that additional tools could be used to explore the correlation between both data.

Chapter 10

Semi-Automatic Myocardial Perfusion Assessment

"If you want to be universal start by painting your own village."

– Leo Tolstoy

Left ventricle functional analysis can be performed using coronary CT angiography (CTA) images as shown in chapter 9. In recent years it has been proposed that CTA imaging might also be used for myocardial perfusion imaging (CTP). During first-pass perfusion of a contrast agent, the perfused myocardium appears slightly brighter (since it is being perfused by blood containing contrast). If a region of the myocardium is hypoperfused (due to coronary occlusion), it will appear darker than the remaining regions.

Myocardial perfusion assessment from these images is performed visually looking for hypoattenuated regions that may correspond to hypoperfusion. Assigning a detected lesion to a specific myocardial segment requires significant effort while the clinician matches the American Heart Association (AHA) standard myocardial segments to the anatomy of each patient. This can also be an additional source of inter-operator variability with different clinicians defining slightly different borders for each segment.

This chapter presents an interactive tool which, based on the existence of a previous left ventricle segmentation, automatically matches the lesions detected by the clinician to a myocardial segment and provides feedback over the image regarding the myocardial segments with detected lesions and their severity.

10.1 Introduction

In recent years it has been proposed that coronary CT angiography (CTA) imaging might also be used for myocardial myocardial CT perfusion (CTP). Image acquisition is performed during first pass perfusion of contrast agent. During this period, perfused myocardium appears slightly brighter (since it is being perfused by blood containing contrast) [116]. If, due to coronary occlusion, a region of the myocardium is hypoperfused, it will appear darker than the remaining regions.

Salerno et al. [117] and Ali et al. [118] consider that combining anatomic (CT) and perfusion imaging (e.g., single-positron emission computed tomography (SPECT) [119]) might be useful for clinicians, although a case-by-case analysis should be performed to avoid unnecessary radiation or cost. In a comparative study by Okada et al. [120], CTP has shown good correlation with myocardial perfusion imaging SPECT for the detection, extent and severity of myocardial perfusion deficits. Thus, combining the anatomical data from CTA with CTP can provide, in a single study not only coronary stenosis assessment but also data on its physiological impact. This is very important since recent studies show that patient prognosis is not improved by revascularization [121] of coronaries presenting moderate stenosis which do not impair blood flow during stress. Furthermore, studies such as the ones presented by Rocha-Filho et al. [122] and Tamarappoo et al. [123] show that a joint CTA/CTP protocol adds value to CTA for coronary stenosis greater than 50%.

In a recent study, Bettencourt et al. [124] show that using CTA and CTP improved multiple detector-row computed tomography (MDCT) accuracy for detection of clinically relevant coronary artery disease (CAD).

Under pharmacological stress, ischemic myocardial areas, which were normally perfused at rest will appear darker (hypoattenuation), allowing detection of reversible ischemia. The detection of these areas, usually related to the presence of significant epicardial coronary stenosis, is one of the main goals of cardiac imaging. Ischemia is related not only to symptoms but also to patient prognosis and its presence (more than detection of coronary plaques or stenosis) should guide treatment. Lesion severity is also an important factor and is typically described as sub-endocardial, when the myocardial wall appears darker near the endocardium, or transmural when the full width of the myocardial wall (from endocardium to epicardium) appears hypoattenuated.

Myocardial perfusion assessment is not a simple task since there is significant inter-patient variability in the way the myocardium is enhanced. Even for a particular patient, significant inter-segment enhancement variation may exist [125, 126]. Furthermore, perfusion analysis is usually performed visually [116] and depends on manual window/level adjustments which might vary among operators [126].

Given that numerous factors contribute for inter and intra-operator variability in myocardial perfusion assessment a semi-automatic (or fully automatic) method is highly desirable. While some efforts are being directed to the development of such methods [126, 127] for quantitative myocardial perfusion assessment, to the best of our knowledge, none of those works considered providing enhanced tools to support the clinician during visual assessment. Visual interpretation of CTP images is still of paramount importance [128], given the common presence of image artefacts [125, 129] which hinder a good performance of automatic methods (among other factors). Introducing interactive tools to support visual perfusion assessment might, for example, reduce the required clinician effort to correctly assign perfusion deficit to a specific myocardial segment, shortening analysis time and improving confidence levels. Furthermore, it can also be an important intermediate step towards automatic assessment methods by enabling systematic validation of their outcomes during the development stage as explained in section 8.5.

A tool is presented, developed using Qt[1], ITK[2], MITK[3] and OpenGl [4] (and integrated in CardioAnalyzer), which supports semi-automatic perfusion assessment from rest and adenosine-induced stress MDCT angiography images. The main innovative features include:

- Perfusion assessment performed by simply identifying hypoperfused regions by clicking on the image;
- Automatic assignment of the hypoperfused regions identified by the clinician to a myocardial segment;
- Visual feedback, presented over the computed tomography (CT) image, depicting the myocardial segments presenting hypoperfusion (as assigned by the clinicians) and the severity of the perfusion deficit;
- Possible interaction with the myocardial perfusion assessment data using a polar map, selecting any segment and overriding the current perfusion assessment, while providing a visual feedback, presented over the image, depicting the corresponding anatomical region (segment);
- Automatic identification of the segment under inspection (identified by where the clinician centres the image view planes) by highlighting it in the polar map;
- Cine perfusion assessment using an animated sequence of the different cardiac phases available to help discard image artefacts.

[1] Qt library: `http://qt.nokia.com/`
[2] Insight Segmentation and Registration Toolkit: `http://www.itk.org`
[3] Medical Imaging Interaction Toolkit: `http://www.mitk.org`
[4] OpenGL library: `http://www.opengl.org`

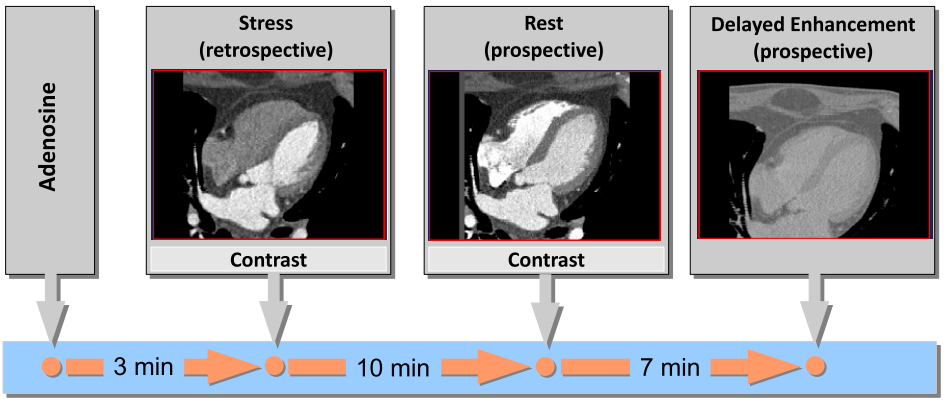


Figure 10.1 – CT perfusion protocol, possibly including additional scan for delayed enhancement assessment.

After a brief description of a typical image acquisition protocol for adenosine-induced stress CTA, used for myocardial perfusion assessment, this chapter presents the usual method for myocardial perfusion assessment. Next, a semi-automatic method for perfusion assessment is presented. After a brief description concerning how it can support the traditional perfusion assessment method, it focuses on how the different features described above allow the identification of the different myocardial segments and associated hypoperfusion severity.

Finally we highlight the main conclusions and several ideas for future work.

10.2 Current Myocardial Perfusion Assessment Protocol

In what follows the myocardial perfusion assessment protocol commonly described in the literature (and used at Centro Hospitalar de Vila Nova de Gaia/Espinho (CHVNG/E)) is presented.

10.2.1 Image acquisition protocol

The image acquisition protocol for CTP is similar to the CTA protocol described in chapter 3. The main difference is that the patient is injected with adenosine, to induce stress, and an additional scan is performed to obtain a cardiac volume with the patient at rest.

Figure 10.1 depicts a typical CTP protocol. After adenosine is administered to the patient, a contrast agent is injected followed by a retrospective acquisition (multi-phase). After waiting a few minutes, with the patient already out of the induced stress, a prospective acquisition (single phase) is performed, also including a contrast agent. Finally, the exam might also include a last prospective acquisition, without a contrast agent, for delayed enhancement assessment.

Images used for this work were obtained by multiple-detector computed tomography (MDCT) using a 64-Slice CT scanner (Somaton Sensation 64, Siemens Medical Solutions, Forchheim,

Germany) with the following scan parameters: gantry rotation time of 330ms; tube voltage of 100kV; tube current of 500 – 700 mAs; electrocardiographic pulsing for decreasing radiation dose with full tube current applied at 60–65% of the cardiac cycle. Patients were under pharmacological stress by a bolus of adenosine delivered according to patient body weight (140 μg/kg/min.).

Given that image acquisition was performed using a low radiation dose protocol – and due to the moving nature of the heart – images tend to present a considerable amount of noise and movement artefacts.

10.2.2 Visual Myocardial Perfusion Assessment

Perfusion assessment is usually performed visually, over one of the cardiac phases available (typically 55%). The clinician examines the image volume using the usual cardiac analysis planes and identifies hypoatennuated regions in the myocardium. If the region is judged to be hypoperfused the clinician identifies perfusion deficit severity and assigns it to a myocardial segment. Assigning a particular lesion to a myocardial segment requires the clinician to mentaly apply the segment division to the current patient's anatomy deciding where each segment ends and the adjacent segments begin. This requires training and considerable effort and might be a source of intra- and inter-clinician variability. Furthermore, when a hypoattenuated region is located, the clinician has to constantly answer the question: is this perfusion deficit already encompassed by any of the segments already marked as presenting perfusion deficit?

This kind of issues might be partially addressed by imposing a systematic approach to image analysis, e.g., using the short-axis plane and navigating from basal to apical segments, but using different view planes allows greater insight over the anatomy and lesions.

The interactive method presented on the following section provides features to address such issues.

10.3 Interactive Tool for Myocardial Perfusion Assessment

One of the main difficulties when performing myocardial perfusion assessment, using the traditional method described above, is the constant need to assign the lesions to a myocardial segment without any support other than the mental model built by the clinician. This can be trained and, with time, a good level of proficiency can be attained. Nevertheless, it is desirable to speed up assessment times and reduce intra- and inter-clinician variability allowing the clinician to focus on detecting the lesions. Therefore, we propose a tool which frees the clinician from explicitly identifying the anatomical location (myocardial segment) of the lesion and allows concentrating solely on identifying the lesions.

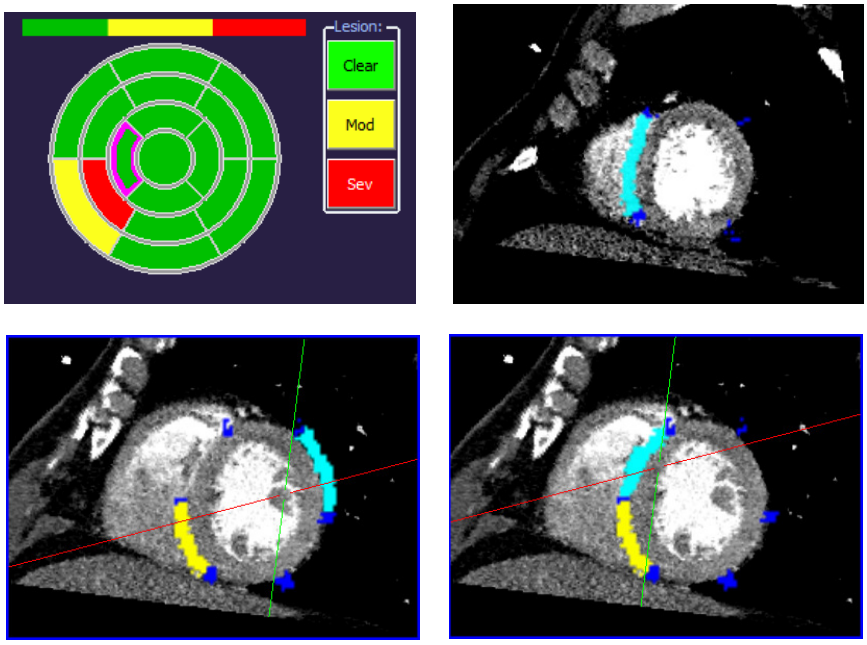


Figure 10.2 – Top left, polar map for myocardial perfusion assessment. Desired segment is selected and perfusion deficit severity (sub-endocardial or transmural) is selected using the buttons on the right. Top right, when a segment is selected on the polar map, the corresponding anatomical region (myocardial segment) is highlighted (in blue). Bottom, when the crossing point of the view planes is moved close to a myocardial segment it will appear highlighted.

10.3.1 Myocardial Segments Definition

As a first step to identify the myocardial segments the myocardium must be segmented. This is performed using the segmentation method described in chapter 5 and available in CardioAnalyzer.

The myocardium segmentation is then processed to define the different myocardial segments [6] as presented in chapter 2: the myocardium segmentation is divided in three parts along its long axis and each of those parts is further divided in six, six and three angular divisions from base to apex. This results in the 17 segments (including the apex) represented in the polar map.

10.3.2 Supporting the Usual Perfusion Assessment Protocol

As a first approach it is possible to perform the usual perfusion assessment protocol by analysing the exam and filling the polar map with the perfusion deficit severity (none, sub-endocardial or transmural).

The clinician simply examines the image using the common views for cardiac analysis: four-chambers, two chambers and short-axis. When a hypoperfused region is detected it is assigned

to a myocardial segment by selecting the corresponding region in the polar map and setting the hypoperfusion level using the appropriate buttons (relevant aspects of the user interface to accomplish such task can be observed on the left of figure 10.2). This kind of analysis is also already supported by a custom software tool routinely used for perfusion assessment from CT at CHVNG/E.

Nevertheless, additional features are supported by our tool. When a region is selected on the polar map, the corresponding anatomical region is highlighted in the image. This is performed by showing a light-blue coloured region on the outside of the myocardial segment as depicted in figure 10.2. Furthermore, if the view planes crossing point is moved close to a myocardial segment, that segment will be highlighted in the image and in the polar map. This is an example of a visualization technique often called interactive linking and brushing [99], i.e., the region under analysis or selected by the user in one visualization is highlighted in the remaining visualizations. Figure 10.2 also shows, on the top right corner, additional small dark blue overlays outside the myocardium. These work as hints for the myocardial segments limits.

10.3.3 Single Cardiac Phase Perfusion Assessment

The selected cardiac phase is loaded and the view planes are automatically set to two-chambers, four-chambers and short-axis views according to the views previously used for segmentation. The window and level are initially set to 300 and 150, respectively, based on experience and on typical values described in the literature (e.g., Valdiviezo et al. [128] and Cury et al. [130]) and considering that common starting values should be used for all exams contributing to similar analysis conditions and criteria. If desired, window and level can be adjusted at any time. Figure 10.3 presents some examples of how different window/level pairs affect image visualization.

As explained earlier, MDCT images present noise, particularly in those cardiac phases where radiation dose is kept to a minimum. Furthermore, CTP images are obtained under stress, which increases noise and motion artefacts. Since myocardial perfusion assessment is to be performed visually, instead of viewing images using a simple cutting plane, a thick multi-planar reconstruction (MPR) is used.

Two options can be adjusted concerning thick MPRs: MPR thickness and MPR computation method (maximum or minimum intensity projection, simple and weighted average voxel value and median voxel level). Typically, clinicians choose the average MPR and a slice thickness around 10mm. Thicker slices, due to larger voxel size, result in decreased image noise and improved contrast for the hypoperfused regions [116].

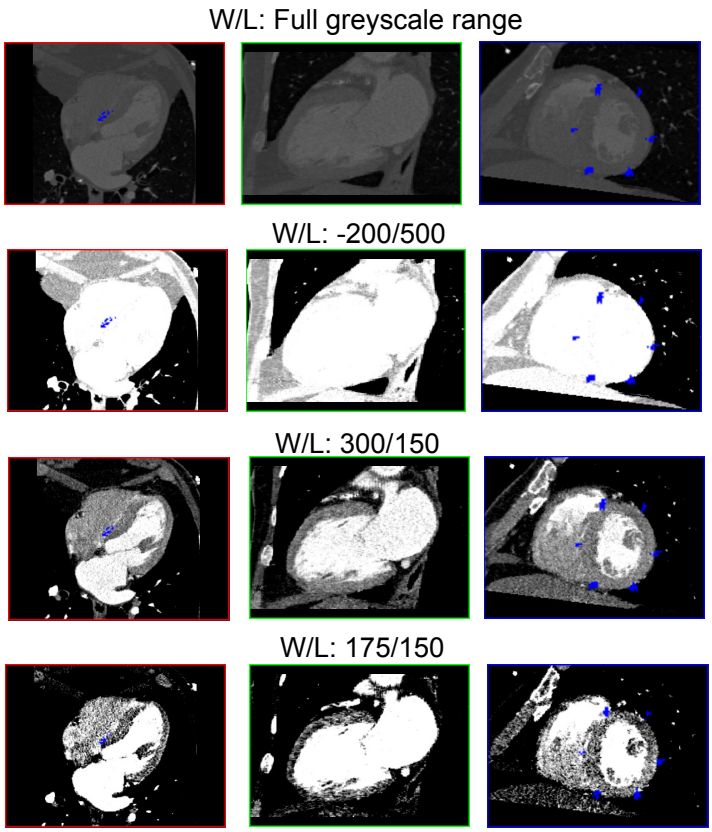


Figure 10.3 – Different window/level (W/L) pairs applied to CT myocardial perfusion image. From top to bottom: no specific W/L applied, W/L usually applied for enhanced lung visualization, W/L=300/150, used as a starting point for myocardial perfusion assessment and W/L=175/150.

Hypoperfusion Identification

The clinician can read the exam, moving the different view planes, looking for hypoperfused regions in the myocardium. To account for a lesion the user clicks over it (more precisely, Shift key + click). Based on its position the corresponding myocardial segment is selected. Point selection has some flexibility since the point does not necessarily need to be precisely put over the myocardium. The corresponding myocardial segment is found based on a proximity rule, i.e., the closest segment to the clicked point is selected.

All segments marked with hypoperfusion keep a visual feedback on the outside by presenting a coloured region starting at the epicardium and extending outside the myocardial segment. This allows the user to easily keep track of which lesions are encompassed by already selected segments and which ones have yet to be identified.

The severity of the lesion is defined by the number of times the user clicks with the mouse inside a segment. One click means a subendocardial hypoperfusion and two stand for a transmural

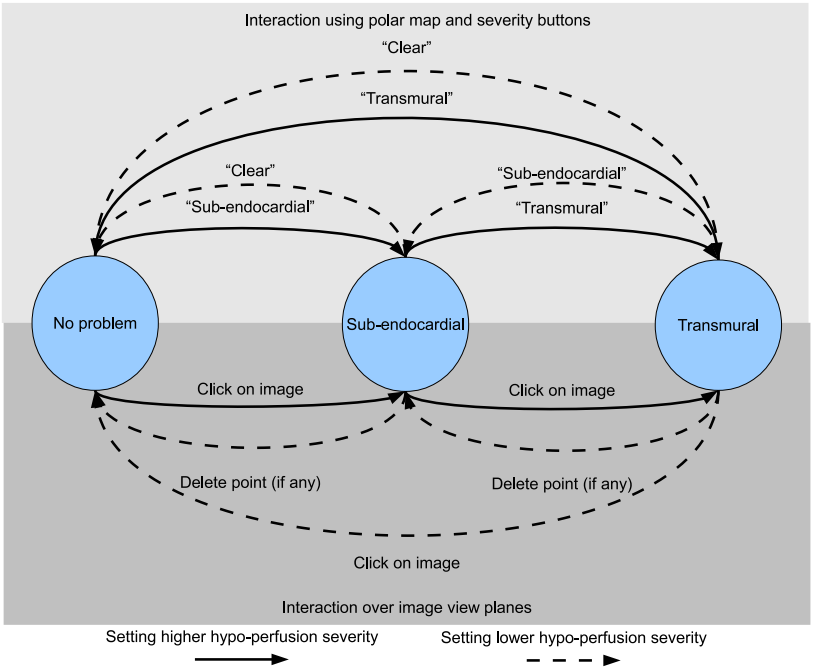


Figure 10.4 – Possible user interactions to set perfusion deficit severity. Two (interchangeable) options are available: interaction using the polar map and severity buttons or clicking over the image.

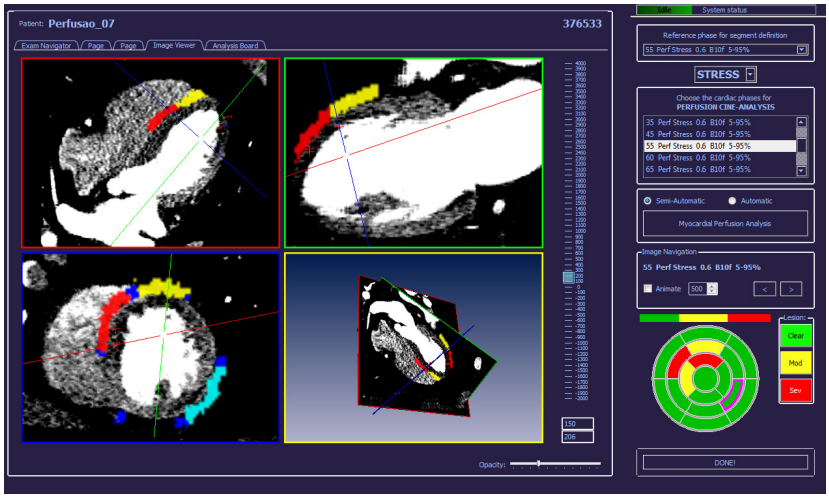


Figure 10.5 – Visual feedback presented for segments considered to be hypoperfused. The visual feedback is present in any of the views. Its colour is directly connected with hypoperfusion severity (yellow). The light blue region seen in the short-axis view highlights the segment which has been selected in the polar map (presenting a magenta outline).

hypoperfusion. It is important to clarify that it is not necessary for the user to identify a lesion multiple times as long as it is already part of a selected segment. Therefore, when the user detects a lesion it can click one time for sub-endocardial hypoperfusion or two times for transmural hypoperfusion and no further action is needed for that segment. It can also happen that after a first analysis, considering a segment as having a moderate lesion, the user, by performing the analysis in a different plane, decides that it should be set to a transmural perfusion deficit. This requires only another mouse click over the region. The position of the earlier mouse click does not limit where the new click can be performed as long as it remains inside the same myocardial segment. Figure 10.4 presents a diagram depicting the different possible interactions to change perfusion deficit severity.

The colour of the visual feedback provided is related with the hypoperfusion severity set for each segment. Figure 10.5 shows visual feedback for two different hypoperfusion levels assigned to different myocardial segments. Please note that the visual feedback can be temporarily disabled and its opacity changed if the user considers it is interfering with myocardium analysis. The polar map is continuously updated according to user interaction performed on the image windows.

Note that both methods (clicking on the lesion or setting severity using the buttons close to the polar map) can be mixed. For example, if the user sets a segment as having a sub-endocardial perfusion deficit using the polar map buttons and later, when analysing the image, decides that it should be set to transmural, a click over the hypoperfused region will set severity to transmural.

Changing lesion severity

If the user wants to change the decision, reducing the severity of the lesion or clearing the selection can be performed in three different ways. The points clicked to set the presence of a hypoperfused region inside a segment are depicted over the image and can be selected and deleted. The visual feedback (and lesion severity) of the corresponding segment will be updated. This is easy to perform if the user is already looking to the added points. On the other hand, if the points have to be located when the user has already moved to different view planes it can be a tiresome task.

A second alternative is to perform additional clicks over the image (and considered segment) until the desired lesion severity is reached. An additional click when the severity is already set to transmural resets that segment back to a "no lesion" decision.

Finally, the user can simply select the desired segment in the polar map and override the decision for that segment. By doing this, all the points that might have been clicked inside the segment will be deleted and the severity set by the user will be considered.

The diagram in figure 10.4 also depicts these three alternatives (in dashed lines).

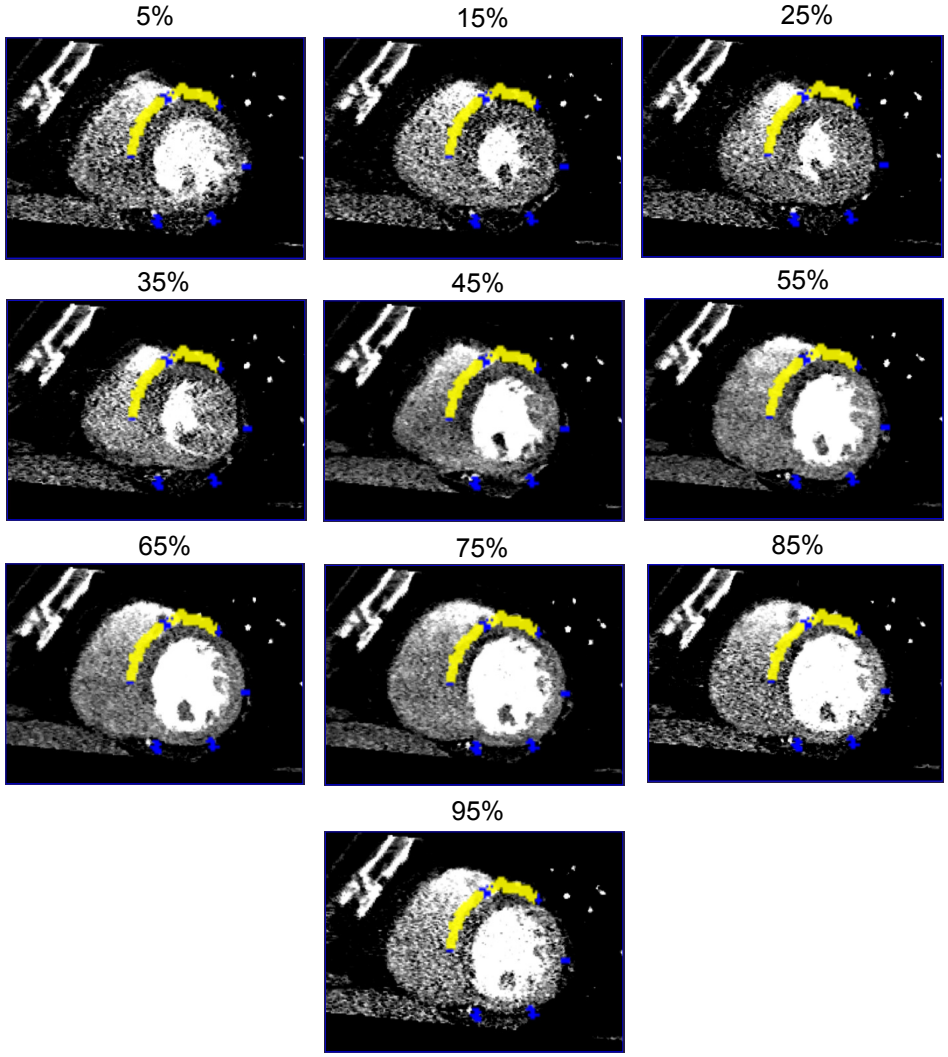


Figure 10.6 – Myocardial perfusion cine-analysis. Clinicians see an animated sequence of the different cardiac phases (in any view plane they find useful) in order to check if a hypoattenuated region is due to a perfusion deficit or an image acquisition artefact. The short-axis left ventricle images sequence presented shows two segments with a perfusion deficit. Notice how the strongly hypoattenuated region keeps the same myocardial position over all phases.

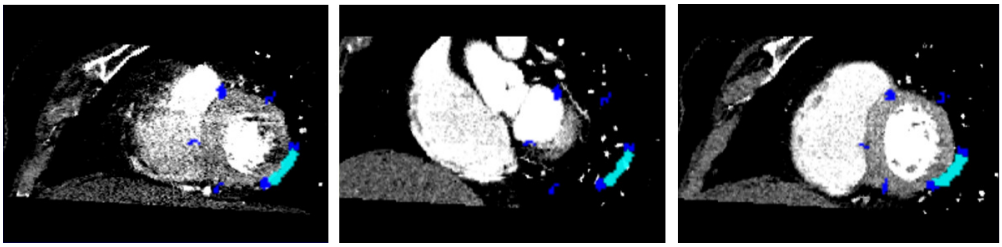


Figure 10.7 – From left to right, short-axis views of cardiac volumes obtained using the described acquisition protocol for myocardial perfusion assessment: 55% during adenosine-induced stress, 55% at rest and 55% at rest after image registration with corresponding cardiac phase during stress.

Myocardial Perfusion Assessment from Rest Images

The features described above are applicable to both stress and rest images. Nevertheless, when considering myocardial perfusion analysis for both cases it should be noted that, since they are acquired at different times, patient position (and defined acquisition volume) will not be the same. Therefore, if left ventricle segmentation has been performed using a stress image (in order to define the myocardial segments) this cannot be used, as is, to define the segments location on the rest image since the left ventricle is in different positions in both images. An alternative would be to perform left ventricle segmentation for the rest image, but this would require an extra segmentation task. The considered cardiac phase is the same for both stress and rest and the only difference is heart location (and probably only slight differences in orientation). Therefore, if both images are registered then the segmentation (myocardial segments location) for the stress phase is also suitable for the rest image.

Figure 10.7 shows a short-axis image for the 55% cardiac phase under stress (left) and the 55% cardiac phase during rest (middle) using the same short-axis plane position for both image volumes. Notice the dark blue hints, for the mycoardial segments borders, present around the myocardium: in the first image they are well positioned and in the middle image they keep their absolute position but since the heart is located differently they are not usable. After registering the stress and rest images the latter is translated and, as can be observed in the rightmost image of figure 10.7, the dark blue hints show proper positioning regarding the myocardium.

Notice that while reading the rest image the clinician can change between the rest and stress images for further insight.

Figure 10.8 – Browsing perfusion data. Users can browse a list of patients and access myocardial perfusion assessment data for rest (left) and stress (right). Between the polar maps a summary of coronary occlusion is presented based on the stress polar map (automatic) or defined by the clinician.

10.3.4 Perfusion Cine-Analysis

Tamarappoo et al. [123] present a perfusion assessment protocol which considers both systole and diastole. The main goal is to rule out the effect of artefacts (usually present in stress CTP images [129, 131]): a region can be hypoattenuated due to the presence of an acquisition artefact and not due to hypoperfusion. For example, a significant and well known cause of such artefacts might be beam hardening and recent works have been presented proposing methods to minimize it [132, 133]. The prospect is that they will be widely available in the near future [129]. Nevertheless, in the meantime, alternative methods need to be introduced to tackle the problem. Using both cardiac phases it is possible to compare the different myocardial segments between both phases. If the hypoattenuation does not appear on both cardiac phases then hypoperfusion can be ruled out.

This method can be further improved by using all cardiac phases available, shown to the user in an animated sequence. While hypoattenuated regions corresponding to hypoperfusion appear throughout the image sequence in an almost constant position, the hypoattenuated regions corresponding to acquisition artefacts tend to drift or disappear. Figure 10.6 shows mid-ventricular short-axis images for all cardiac phases available. Notice that for the segments with a subendocardial hypoperfusion (yellow feedback on the outside of the myocardium) the hypoattenuated

regions can be detected in almost all cardiac phases, in the same myocardial position, thus ruling out the chance of an acquisition artefact.

As with the single-phase assessment the user just needs to click (Shift key + click) over the hypoperfused regions. The corresponding myocardial segments will be automatically identified and the polar map filled accordingly. Since there is left ventricle position coherence along the cardiac cycle only one myocardium segmentation is needed (preferably a diastolic phase) to ensure proper identification of the myocardial segments.

While the image sequence is shown the user can freely interact with the images, changing the view planes or setting different animation parameters such as animation speed. Image sequence animation can be stopped at any time for detailed inspection of any cardiac phase and navigation through the sequence can be performed manually.

Furthermore, showing the different cardiac phases in an animated sequence might allow the visual detection of motion abnormalities [116], which increases confidence levels when assessing myocardial perfusion: a region for which the clinician is not sure of hypoperfusion, if contractility problems are detected, they provide extra data to decide positively.

10.3.5 Browsing Perfusion Assessment Data

Perfusion assessment data is stored and can be accessed without re-opening the exam. When a patient is selected from the list the polar map for myocardial perfusion in stress (and rest if available) is presented (figure 10.8). An interpretation of the polar map is proposed showing which coronary arteries might present occlusion. This is still a very unsophisticated method based on the general assignment of myocardial segments to the coronary arteries [6], as shown in figure 10.9, but has been used in the literature for similar purposes [130]. This automatic interpretation might fail depending, for example, on which vessel is dominant (and supplies inferior and inferoseptal territories [116]), and is patient specific [134]. Nevertheless, it can be corrected/changed by the clinician at any time.

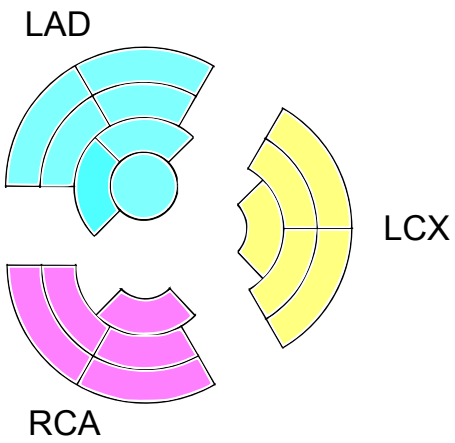


Figure 10.9 – Typicial assignment of myocardial segments to coronary supply territories [6]: left anterior descending (LAD), right coronary artery (RCA) and circumflex coronary artery (LCX).

If the clinician chooses to repeat the analysis it will have to do it from scratch, i.e., the previous analysis will not be used as a starting point. We got clinician feedback regarding this issue stating that if there is a need to repeat an assessment then it is preferable to do it without

any data that can somehow influence judgement.

10.4 User Feedback

To get user feedback regarding the proposed tool, we asked four clinicians for their opinion about the tool and its features. A clinician was also asked to use it to perform myocardial perfusion assessment for 20 exams in order to force using the tool in a "real scenario". Our main purpose was to understand if the tool was adequate for the envisaged task and if any adjustments were necessary.

Clinicians reacted very positively to the tool and had no problems understanding how it worked. While using it, the clinicians provided several suggestions for improvement. Regarding interaction with the different view planes, it was possible to use keyboard shortcuts to move the plane up and down along its normal vector direction. To choose which plane was to be affected by the keyboard shortcuts the user had to click over the corresponding window. Unfortunately, this also affects the position of the crossing point for the view planes and therefore the different view planes were moved. In order to solve this problem our first approach was to automatically choose the window under the mouse cursor. This, we consider, would provide a very simple and intuitive window selection method. Unfortunately, implementing this solution was not technically possible and an alternative was provided by using the right mouse click (as opposed to the left mouse click to move the view planes crossing point).

The visual feedback for the segments closest to view plane crossing and the visual marks depicting segment limits were initially always present. This was sometimes considered intrusive and not very useful. As a result, these features were set as disabled, by default, and an option was added to enable them if necessary.

In order to get qualitative data regarding image visualization and the effects of thick slice options a clinician was asked to open the same exam both in our tool and the workstation typically used for myocardial perfusion assessment and compare them. The clinician stated that he did not notice any difference between the two images.

One aspect in which our tool differed from the image visualization workstation typically used for myocardial perfusion assessment concerned the time taken from patient selection to image availability. In our tool this time was larger since, while in the usual workstation the image could start being examined after just one cardiac phase was loaded, in our tool all phases had to be loaded before any interaction could take place.

10.5 Conclusions

This chapter presents an interactive tool for myocardial perfusion assessment from CTP images. Its main goal is to provide clinicians an easier way to evaluate myocardial perfusion by allowing them to focus solely on detecting the lesions while their anatomical location (myocardial segment) is automatically set based on a previous left ventricle (LV) segmentation.

The tool was well received by clinicians and preliminary validation has provided a set of suggestions for improvements which have been carried out striving to improve interactivity and visual feedback.

One important improvement that should be considered concerns the time needed from patient loading until the exam is available for analysis. One possible approach to tackle this issue is to allow parallel image loading, i.e., while the already loaded cardiac phases can be examined, the remaining phases are loaded in the background.

This tool is also important for the development of automatic myocardial perfusion assessment methods. Some variability is prone to occur, from clinician to clinician, regarding the myocardial segments anatomical location. If the automatic method considers slightly different segments (e.g., by considering a smaller distance from apex to the basal region), validation of automatic data versus visual assessment data, on a segment to segment basis, might be difficult. A more general validation can be performed, considering just the coronary supply territories, but segmental data might be important during development and method validation to provide a clue to which segments should be identified as hypoperfused. By using this interactive tool the same myocardium segmentation (and, consequently, myocardial segments) can be used for visual and automatic perfusion assessment.

Part IV

Conclusions

Chapter 11

Conclusions and Further Work

> "Geralmente, quando se trabalha numa tese só se pensa no momento em que ela estará terminada: sonha-se com as férias que se seguirão. Mas se o trabalho for bem feito, normalmente, depois da tese, verificar-se-á a irrupção de um grande frenesim de trabalho. Deseja-se aprofundar os pontos negligenciados.[...] É isto sinal de que a tese vos activou o metabolismo intelectual [...] e que são agora vítimas de uma coacção para investigar, um pouco como o Chaplin dos Tempos Modernos, que continuava a apertar parafusos mesmo depois do trabalho."
>
> – Umberto Eco, Como Si Fa Una Tesi Di Laurea, 1977

This chapter presents a summary of the main contributions of the work carried concerning LV functional analysis from coronary CT angiography. These include a new semi-automatic segmentation method, validated by experienced radiographers, a tool for visual exploration of different parameters characterizing LV function over multiple cardiac phases and an interactive tool for visual assessment of myocardial perfusion.

As important as the work carried out is its potential to provide grounds for improvements and future developments. In that sense, this chapter also presents some ideas for future work.

11.1 Overall Conclusions

Non-invasive cardiac angiography using MDCT scanners is, nowadays, routinely used in clinical scenarios. The amount of data provided along the cardiac cycle and increasing new applications, e.g., myocardial perfusion assessment make it an important focus point for research.

In the work presented in this thesis our main goal was to explore the available data and provide ways of enhancing/easing user experience while reading the exams for diagnosis purposes.

The first step was to develop a segmentation tool which allowed obtaining the data required for analysis. This was performed taking in consideration that multiple cardiac phases were to be processed. An editing tool was also proposed which allowed 3D editing in any view plane. This provided that editing, when needed, could be performed faster than with a 2D tool, which would require a slice by slice editing. On the other hand, this editing tool requires a different approach to editing given that, since the tool is 3D, looking solely to one viewplane is not enough to have a full idea of the regions affected. This difference towards the typical 2D editing tools available had, as far as we could understand, an influence in confidence levels when using the tool and, therefore, in the overall satisfaction scores obtained during evaluation.

The evaluation method used consisted in asking three experienced radiographers to segment the LV for six cardiac phases and for four different patients. The radiographers were asked to repeat the segmentations at a later time for intra-radiographer variability assessment. The obtained segmentations were compared using the Jaccard coefficient and voxelwise difference. Both metrics were selected from a wider set of metrics using a comparative study to exclude metrics which provided redundant data. Comparison data has shown very good results concerning inter- and intra-radiographer variability which enables the proposed tool to be used to obtain data for analysis.

Having segmented LV data for multiple cardiac phases allowed the development of an analysis tool providing several parameters characterizing global and regional left ventricle function. The data provided by the computed parameters is shown using a set of visualizations allowing synchronized visual exploration of the different data. The main purpose is to provide means for clinicians to explore the data and gather insight over their meaning and their correlation with each other and with diagnosis outcomes. To the best of our knowledge this is the first tool proposed based on multiple cardiac phases of a CTA exam and the only system described in the literature providing a wide range of parameters supported by different visualization and analysis features.

Finally, an interactive tool for myocardial perfusion assessment was proposed which allows clinicians to visually assess myocardial perfusion without having to constantly perform the assignment of each detected lesion to a myocardial segment, according to the myocardium analysis standards. Instead, the proposed tool automatically assigns the lesions to the corresponding my-

ocardial segment. Furthermore, rest images can be more easily compared with stress images due to image registration between the two.

11.2 Future Work

Some ideas for improvement of the presented work have been provided along this document in the conclusions of each chapter. Nevertheless, additional ideas can still be considered.

Segmentation can be improved in different ways. Based on the results obtained during evaluation, radiographers performed a small amount of editing on the segmentations of the different cardiac phases (excluding the reference phase). Therefore, in order to speed segmentation the step concerning user revision of segmentations for cardiac phases other than the reference phase might be dropped;

Additional parameters characterizing LV function can be tested, with a particular emphasis in myocardial perfusion assessment. The data collected using the interactive tool for myocardial perfusion assessment may be used for systematic validation of such parameters since it provides a common ground for comparison. This is due to the fact that the segmentation used for automatically defining the myocardial segments in the interactive tool can also be used to compute myocardial perfusion assessment parameters. During validation this ensures that the segment the clinician classified as having a perfusion deficit corresponds exactly to the same anatomical region covered by the segment that is being automatically assessed.

Still concerning the interactive tool for myocardial perfusion assessment, image loading should be improved by allowing user interaction immediately after the first cardiac phase has been loaded. This was one of the features suggested by clinicians and can speed analysis since it allows image inspection much sooner.

Integration with PACS is essential to ensure proper integration of the proposed tools in the clinicians/radiographers work flow. At this moment to analyse an exam with the proposed tools it has to be exported to DICOM, using an existing workstation with PACS access. Even though this takes only a couple of minutes and might even be performed, for example, during the night, if the exams are readily available for analysis the proposed tools will potentially be used on a regular basis as part of the exam reading tasks.

Routine use of the tools and methods presented in this document would be a very important (and desirable) conclusion of the work presented in this document. It would provide further evidence of their relevance and an interesting and challenging scenario for further improvements and developments.

Appendices

Appendix A

Tools for Medical Image Processing, Visualization and Analysis

Software tools for medical image processing, visualization and analysis play a very important role in medical scenarios helping practitioners, during the diagnosis process, to take the most out of data provided by modern devices such as MDCT scanners. Such importance is mirrored by a strong research activity with constant developments including new image processing algorithms and visualization techniques for specific purposes, e.g., functional heart analysis. Developing such tools requires a large amount of work involving several stages such as, reading image data, processing, visualization, manipulation and analysis support, and saving the data.

Several free and open-source software tools exist which might help users in this development process but they are often presented (and reviewed) as individual tools. Since they have different features they might be suited for different development stages and their integration with each other, if possible, might help improve the development pipeline while providing the desirable shorter distance between prototyping (at the research level) and end-user applications (at clinical level).

Based on an analysis performed to different tools, their features and their common application scenarios, suggestions are provided regarding their use at different development stages.

A.1 Introduction

Medical image processing, visualization and analysis systems are nowadays a common and indispensable tool for practitioners helping during the diagnosis process to properly explore and analyse the huge amounts of data provided by modern diagnosis imaging devices.

Constant scientific developments lead to increasingly accurate devices and improved analysis methods, but developing a new system for medical data processing and visualization can be a tiresome task as it involves time consuming steps, such as new algorithms implementation and user interface development, which add a very significant overhead to the process. Often, when given a new dataset, the developer starts by testing processing methods already described in the literature and then, if they do not provide the desired results, develops and tests his own methods which always requires a suitable environment to visualize and analyse these preliminary results.

Along the past decade a considerable effort has been made in building tools and providing libraries which help users to more rapidly build software applications for medical imaging. Even though some of those tools require license fees (such as Analyse[1]), a large number is provided free of charge and users might even have access to their source code, thus encouraging use and allowing validation and constant enhancements by the scientific community [135] [136].

Recent studies have presented some of the tools available for medical image processing and visualization. These tend to be more descriptive [137][138], comparative but from an experienced user point of view [139] or focusing more on image loading, viewing and saving capabilities and less on the processing features [140]. Meanwhile, new tools are constantly being presented and there is a need to place them comparatively to the existing ones.

An important aspect not covered in the literature is how all these tools, instead of being seen as individual tools, might be used together in a development pipeline from prototyping to an end-user application. Is it possible to use one tool for prototyping and then continue work using a different, more suited tool, for the later stages?

The purpose of this appendix is to review several of those tools but from the viewpoint of a less experienced user starting (or wanting to evolve) in medical image processing and analysis. In this scenario it is important to address issues such as the existence of user documentation and the number of native features provided by each tool. Although many of them allow user extensions, it is far better to be able to avoid this task as much as possible. Writing and, specially, testing new algorithms for medical applications requires careful methods and a lot of time [135].

We aim to provide information addressing what we think are the main concerns when one needs to choose a medical image visualization and processing tool:

[1] Analyze: `http://www.mayo.edu/bir/Software/Analyze/Analyze.html`

- What kind of features are natively provided?
- What application areas is it used in?
- Does it support the processing/visualization tools I need?
- Is there support documentation?
- I am already acquainted with a certain software library. Will I be able to use that knowledge with the new tool?

If the prospect is to evolve and develop or integrate new features, these aspects should also be considered:

- Is the source code available so I can modify it or use parts in my work?
- Can it be easily expanded with features that I develop?
- What tool should I start with to perform prototyping, if I plan to move to a more complex framework at a later development stage?

This appendix starts by presenting several libraries/toolkits suited for medical image processing, visualization and manipulation followed by a description of the main features of nine complete software tools for the same purpose, distinguishing between integrated environments and application builders. A short discussion follows comparing the advantages and disadvantages of each tool. Finally, some conclusions are presented.

A.2 Toolkits

This section presents a set of libraries/toolkits that can be used as building blocks of software applications covering image processing, visualization and user interface design.

The criteria used to select the presented libraries concerned their popularity and the range of features provided trying to cover all stages of the development process (processing, visualization, user interface) and notable areas of interest as image-guided surgery.

A.2.1 ITK – Insight Segmentation & Registration Toolkit

The Insight Segmentation & Registration Toolkit [141] [142] is a joint project of several institutions and funded by the National Library of Medicine. It consists of an open-source toolkit, developed in C++, for image processing, segmentation and registration. These operations can be performed

by connecting several filters in a data flow pipeline. Each filter stores its state and will only be executed when necessary. It provides data readers and writers supporting several image formats, including DICOM, and multi-dimensional data through template instantiation. ITK also supports multi-threading.

A major advantage is the fact that ITK provides a large number of algorithms, already tested by the community, including thresholding, edge detection, smoothing, region growing, level-sets and multimodal registrations to name just a few. A very useful user's guide with an explanation of the main features and a large amount of examples is available.

ITK provides wrappers for interpreted languages (e.g., Tcl and Python), and interfaces to the Visualization Toolkit (VTK).

Image visualization (some simple viewers are provided, as an example, built using FLTK[2] and Qt[3]), manipulation or analysis features are not provided by ITK and so it will eventually have to be used with other libraries to cover those aspects.

This library's importance is well depicted in Medical Image Analysis' special issue about ITK [143].

Several tools have appeared which try to ease ITK usage (especially for those with low or no programming skills) by providing a graphical user interface where the processing pipeline can be built. Apart from some of the tools analysed in section A.3 there are two other small tools for this purpose: itkFlowRun [144] and itkBoard [145].

A.2.2 VTK – Visualization Toolkit

The Visualization Toolkit (VTK), developed at Kitware, Inc. is a free open-source library, developed in C++, which provides a large number of visual data representations, along with several low-level interaction support such as picking, rotating and scaling.

Although primarily a visualization toolkit, VTK also supports a few 2D/3D image processing and manipulation features. It provides a data type for dealing with image data (including the DICOM format) and several filters to accomplish tasks such as smoothing, re-slicing or volume of interest extraction.

For fast prototyping, VTK provides bindings for Tcl and Python.

A.2.3 MITK – Medical Imaging Interaction Toolkit

ITK provides a rich set of image processing tools but its application in clinical scenarios requires additional visualization and interaction features. Eventhough in several situations visualization

[2]FLTK: `www.fltk.org/`
[3]Qt: `qt.nokia.com/products/`

can be handled by VTK it still lacks extended interactions support.

The Medical Imaging Interaction Toolkit (MITK)[4] [146][147], an open-source library written in C++ and developed at the German Cancer Research Center, Division of Medical and Biological Informatics, Heidelberg, is presented as an extension to ITK and VTK with the main purpose of reducing the effort required to develop interactive applications for medical image analysis by easing the combination between these two libraries. It also provides increased support for complex interactions and visualization options including different (consistent) views of the same data (including multi planar reconstructions) or $3D + t$ datasets visualization. Image data, resulting segmentations and surfaces are managed through a data tree whose branches visibility can be toggled. Another interesting feature provided is an undo option.

Included in MITK's source code is a software application (known as MainApp) which gathers the main features supported by the toolkit and a set of tools (e.g., interactive segmentation, editing, volume rendering) for testing purposes. It allows a jump-start for anyone wanting to use the toolkit. It is possible to use this software application as the main structure of our software application and develop additional features.

Since MITK is derived from ITK and VTK, it presents a similar architecture and therefore, developers used to ITK or VTK will find many similarities.

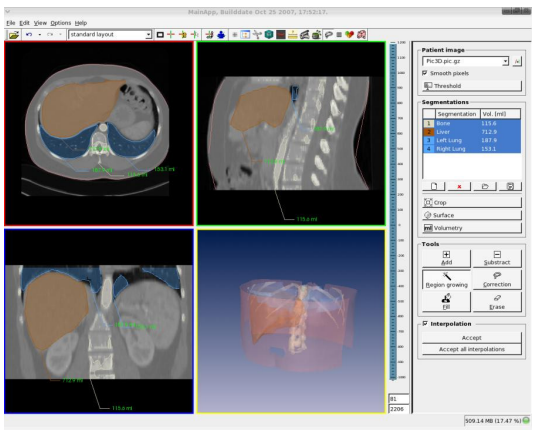


Figure A.1 – Example showing some of the features provided by an user interface designed using MITK.

Figure A.1 shows an example of a software application developed using MITK. An image volume is visualized with several segmented regions using three orthogonal views and volume rendering.

[4] MITK: `http://www.mitk.org`

A.2.4 MITK – Medical Imaging Toolkit

The Medical Imaging Toolkit (also known as MITK) [148][5] is an open-source library developed in C++ at the Medical Image Processing Group, Institute of Automation, Chinese Academy of Sciences. It is presented as an alternative to VTK and ITK and as a more consistent, simple and intuitive to use toolkit, since it is not developed using generic programming (templates, etc.). Even though inspired in VTK and ITK this library does not use any code from them. It follows the data flow model and allows reading and writing a large number of image formats including DICOM. MITK provides a large set of algorithms for smoothing, segmentation and registration along with surface reconstruction and volume rendering. Data set processing and visualization can be performed out-of-core.

MITK also allows managing interaction and the development of graphical user interfaces. The same group has used MITK to develop 3DMed, a free application for medical imaging processing, analysis and visualization, supporting segmentation and registration algorithms among other features.

A.2.5 IGSTK – Image-Guided Surgery Toolkit

Interventional radiology suites have, nowadays, an important role in surgical scenarios. Apart from the image processing, visualization and analysis features provided by the previously mentioned toolkits it is also necessary to have specific tools to, e.g., deal with external data coming from trackers associated with the surgical instruments or manage a suitable and common coordinate system integrating that information with anatomical data. It is also important to present it to the practitioner in the most suitable way possible.

The Image-Guided Surgery Toolkit [149] [150][6] is a free and open-source toolkit, written in C++, developed by the Insight Software Consortium. It is built using, among other libraries, ITK and VTK, for image processing and visualization, and FLTK for the user interfaces. It provides a set of modules which allow dealing with tracker information, object representation and visualization. The provided documentation covers a wide range of aspects concerning development using IGSTK.

Figure A.2 presents the user interface of a software application developed using IGSTK for robot-assisted needle placement in biopsies which helps planning the needle path to reach the tumour while avoiding other structures such as the ribs.

[5] Medical Imaging Toolkit: `http://www.mitk.net`
[6] IGSTK – Image Guided Surgery Toolkit: `http://www.igstk.org`

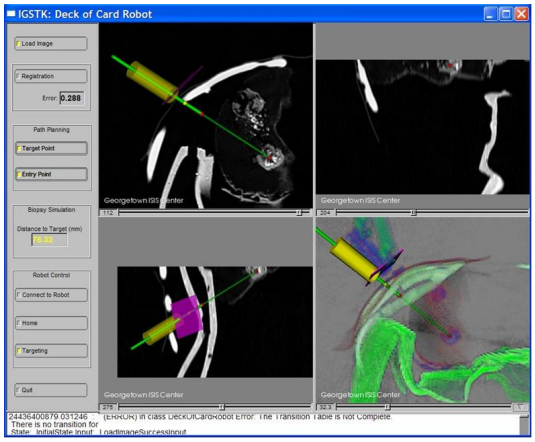


Figure A.2 – User interface of an application developed using IGSTK for robot-assisted needle placement in biopsies [149].

A.2.6 MARVIN

MARVIN [151][7] is an open-source library developed in C++ at MEM Research Center, University of Bern, which provides a framework for fast development of software applications to support biomedical and clinical research.

Computer-assisted surgery is one of the major application areas in which MARVIN is used and the support for several devices is provided through plugins.

For visualization, MARVIN uses Coin3D[8], an implementation of Open Inventor, but alternative visualization libraries can be used due to complete independence between the data and visualization layers.

MARVIN uses Qt for the graphical user interface.

A.2.7 User-interface Development

There are additional open-source, platform independent libraries for user interface development such as wxWidgets[9], Fast Light Toolkit (FLTK[10]), and KWWidgets[11]. Either of them has been successfully used to build medical imaging applications and provides similar features. Nevertheless, if users intend to use VTK for visualization it is important to note that KWWidgets provides a set of high-level widgets suited to easily configure visualization parameters. A notable example is that of transfer function edition which can be accomplished by the widget presented in figure A.3.

[7] MARVIN: `https://choroidea.unibe.ch/marvin`
[8] Coin3D – 3D graphics developer kit: `http://coin3d.org`
[9] wxWidgets: `http://wxwidgets.org`
[10] Fast Light Toolkit (FLTK): `http://www.fltk.org/`
[11] KWWidgets: `htp://kwwidgets.org`

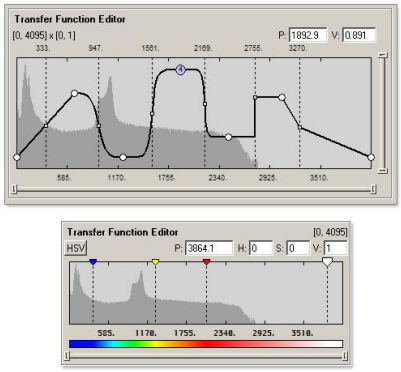


Figure A.3 – Example of one of the high-level widgets provided by KWWidgets, suited for visualization parameters configuration, e.g., in VTK.

A.3 Software Applications

This section presents ten free and open-source frameworks for medical image processing, visualization and analysis. The criteria considered for selection included how easily these tools could be used for medical imaging. There are several generic tools available that can be used for that purpose (and are mentioned in a recent survey [139]) but require some adaptations and do not provide such a "natural" environment (e.g., Paraview[12]). Another criterion used was that they should include image processing capabilities and not only visualization. Examples of interesting tools which fall in the latter category are DicomWorks[13] and Syngo FastView[14]. Attention was concentrated on tools with a considerable number of users and subsequently more testing "under fire" but also on some which have been recently presented and seem to include a promising set of features and potential for further development.

Paying attention to the importance and widespread use of ITK, a large majority of the analysed tools integrates (to different extents) functionality provided by that library. Furthermore, they can all be run on Windows, Linux and MacOS.

Regarding the software tools one should consider the distinction between the integrated environments, i.e., those tools which provide loading an image data set and then processing and visualization using methods available from menus or toolbars; and data flow tools (also known as application builders) which allow building an image processing and visualization pipeline by graphically connecting different blocks. While the first are more suitable for end-users which need to explore image data sets and apply a small number of processing methods (e.g., noise removal), the latter are suited for fast prototyping and testing given the fact that it is easy to insert new

[12]Paraview: `http://www.paraview.org`
[13]DicomWorks: `http://dicom.online.fr`
[14]Syngo Fast View: `http://www.medical.siemens.com`

modules in the network and connect several processing operations to be applied to a data set.

A.3.1 Integrated Environments

Five tools for medical image analysis, visualization and processing are presented in the following sub-sections.

3D Slicer

3D Slicer[15][152][153] is a free, open-source software application, developed by Steve Pieper and several other contributors, for visualization and analysis of medical images. It provides interactive visualization of 3D images, editing, segmentation, registration (ITK) and visualization (VTK) of tracking information for image-guided procedures (using IGSTK).

3D Slicer functionality can also be extended by developing new modules. The Execution Model allows including command line applications as modules by specifying the new module's inputs, outputs and parameters using a XML file and then implementing the desired functionality in C++ using, for example, ITK. This might also get handy to test existing implementations in an environment already suited for medical image visualization and analysis.

3D Slicer user interface (figure A.4) has been developed using KWWidgets.

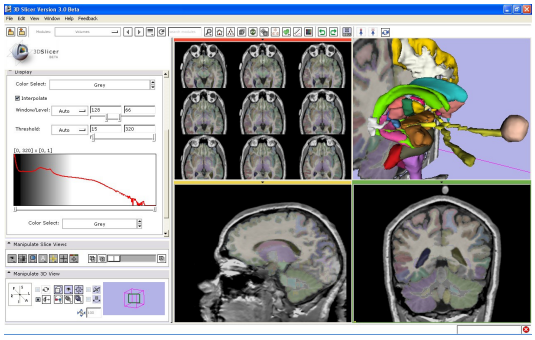


Figure A.4 – 3D Slicer's user interface showing several windows for head image data visualization.

AMIDE – A Medical Imaging Data Examiner

AMIDE[16] [154][155] is a free, open-source software tool developed in C++ by Andreas Loening. It supports a wide range of image formats including DICOM and provides some image processing algorithms such as anisotropic filtering and thresholding, along with different visualization options

[15] 3D Slicer: `http://www.slicer.org`
[16] AMIDE – Medical Data Examiner: `http://amide.sourceforge.net`

(see figure A.5 for some examples) including arbitrary orientations of the data set, multiple data set fusion/overlay and colour map selection. Region and volume of interest definition are also supported and can be saved in XML format. AMIDE allows aligning data sets using fiducial markers.

Volume rendering is also provided and supports several data sets at the same time.

AMIDE does not provide the possibility of extending its functionality through plugins but it is possible to directly change its source code to accommodate user required modifications.

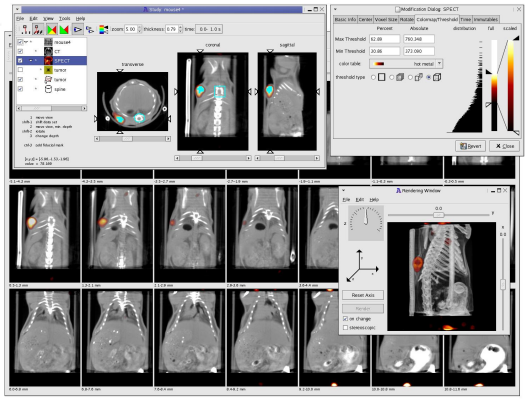


Figure A.5 – User interface of AMIDE showing multiple views of a 3D image set, including multiple adjacent slices.

BioImage Suite

Developed at Yale University, BioImage Suite[17] [156] is an open source tool, developed in C++ and Tcl, which integrates a set of different software applications for image visualization, analysis and processing in general but also specifically suited modules for different application domains including brain imaging and cardiac studies. It integrates features from both ITK and VTK.

MIPAV – Medical Image Processing, Analysis, and Visualization

MIPAV[18] is a free Java software application for medical image processing, analysis and visualization (figure A.6) developed at the Center for Information Technology, USA. It supports reading and writing a wide range of formats, including DICOM, and provides several image processing algorithms (several of those provided by ITK). It has an extensive set of volume of interest definition options including one supported by live-wire techniques. Volume rendering is also available and a GPU-based approach is provided.

[17] BioImage Suite: `http://www.bioimagesuite.org`
[18] MIPAV: `http://mipav.cit.nih.gov`

MIPAV supports the inclusion of three different types of plugins written in Java (algorithms, data type support and visualization), allowing users to expand this tool's capabilities according to their needs.

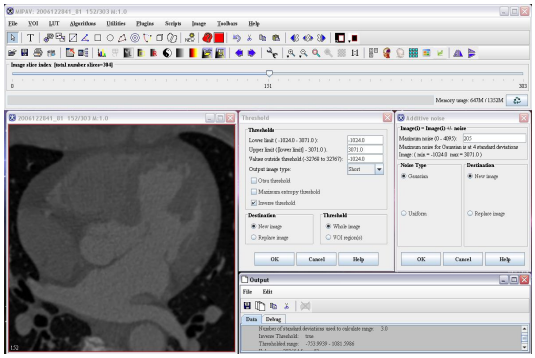


Figure A.6 – MIPAV's user interface showing several of the available tool windows.

MedINRIA

Developed at the INRIA institute, MedINRIA[19] [157] is a set of software tools, gathered inside the same environment, for image processing, visualization and analysis. It is developed in C++ and uses ITK, for image processing, VTK, for visualization, and wxWidgets for user interface design. As stated by the authors, this tool is aimed for clinicians. This might explain why no source code is provided nor any indication on methods which can be used to add user defined features. At this moment, the provided applications, apart from the Image Viewer, seem to focus on brain imaging allowing diffusion tensor magnetic resonance imaging (DTMRI) analysis, tensor visualization and semi-automatic segmentation of multiple sclerosis lesions. A registration tool is also described but was not available in the tested version.

A tool named SepINRIA[20] has been presented which was developed based on the MedINRIA framework and focuses solely on multiple sclerosis brain MRI analysis.

ImageJ

Initially developed by Wayne Rasband at the National Institutes of Health (NIH), ImageJ[21] is an open source software tool developed in Java. It can handle a large variety of image formats, including DICOM, and has a plugin architecture. Several plugins are available to support numerous

[19] MedINRIA: `www-sop.inria.fr/asclepios/software/MedINRIA/`
[20] SepINRIA: `www-sop.inria.fr/asclepios/software/SepINRIA/`
[21] ImageJ – Image Processing and Analysis in Java: `rsbweb.nih.gov/ij/`

features such as smoothing, thresholding, registration or volume rendering. It also allows defining and saving regions-of-interest (ROIs).

A very interesting feature of ImageJ is that it supports command macros, i.e., storing a sequence of processing operations which can then be easily applied to different images. Batch processing is also possible which can get handy, for example, when there is a need to apply a set of pre-processing steps to a large number of images.

A.3.2 Application Builders

Recent years have seen the development of application builders for different application areas. The following sub-sections present five notable examples of such tools supporting medical image analysis, processing and visualization.

MeVisLab

MeVisLab[22] [158] is a tool, developed and used by Mevis Medical Solutions AG and Fraunhofer MEVIS, Bremen, Germany, which allows building an image processing/visualization pipeline by connecting several modules using a graphical interface (see Figure A.7 for a module network example). These modules consist in several image processing methods (such as resampling, blurring and thresholding) and visualization tools along with modules which allow loading and saving image files. DICOM files need to be imported to MeVisLab format so they can be processed which can be easily done with one of the available modules. MeVisLab includes several modules providing segmentation, registration and quantitative morphological and functional analysis. A powerful feature is that it integrates the well known libraries VTK and ITK as network modules allowing fast prototyping using the potential of those toolkits without need of knowing a programming language. Figure A.7 (on the right) shows a sample processing network built in MeVisLab. An image is loaded (imgLoad), anonymized (AnonymizeMacro2) and then a sub-image is extracted (subImage) to consider, for example, just some of the slices. Gaussian filtering is applied (itkDiscreteGaussianFilter) and a SynchroView2D module is used to compare the original and filtered image side-by-side (as seen in Figure A.7). Finally, the image is saved (imgSave).

Another feature is that it allows gathering a complex module network in a macro module (with user definable inputs and outputs) thus contributing for a greater understanding of really large networks. Each module's behaviour can be configured using a panel containing adjustable parameters.

MeVisLab also allows building simple graphical user interfaces associated with particular module networks by placing chosen elements of each module individual panels in a single user interface

[22] MeVisLab: `www.mevislab.de/`

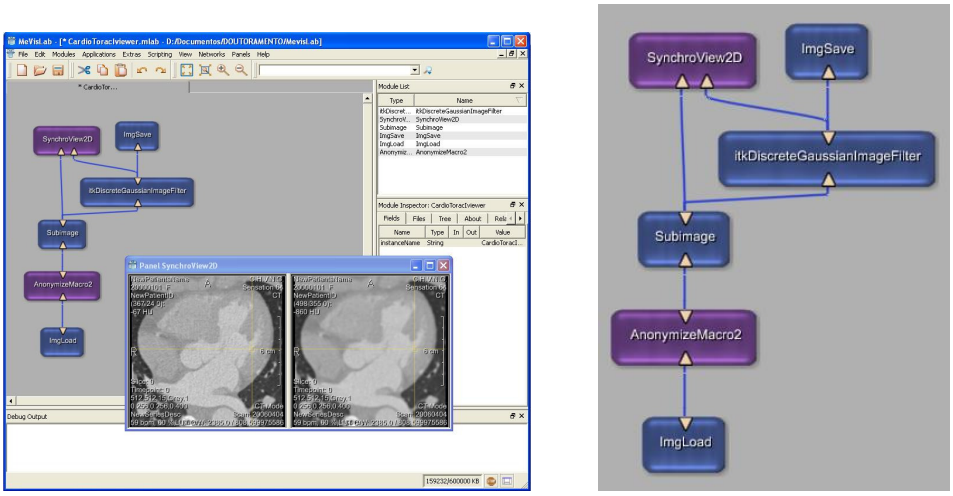


Figure A.7 – MeVisLab environment (left) and detailed view of a sample network (right) built with some of the available modules.

panel. It is also possible to develop scripts (e.g., in Java) and place buttons in the interface which will trigger their execution. An example graphical user interface built for a specific module network is presented in Figure A.8.

Figure A.8 – Graphical user interface built with MeVisLab showing the axial view of a heart exam and several adjustable parameters concerning left ventricle segmentation.

New modules can be developed and integrated in MeVisLab by using a built-in wizard and then filling the gaps in the pre-built code structure (in C++). This is very useful to cope with those situations when a very specific processing operation is desired and not supported by any of the existing modules.

MeVisLab provides a reasonable documentation and a very interesting feature is that the majority of modules available in MeVisLab has an associated example network which shows how that module can be used in a typical situation.

Recently, an add-on package called Medical Exploration Toolkit (METK) has been pre-

sented[23] [159] which provides a set of modules for advanced visualization and exploration. This add-on aims to help on the development of complex applications with a reduced number of modules and an improved connectivity between them.

SciRun

SciRun[24] is a processing, visualization and analysis package, developed at the Scientific Computing and Imaging Institute, University of Utah, which can be used in a wide variety of applications [160]. It provides building an image processing and visualization pipeline by connecting different modules in a network using a graphical user interface. These modules include several ITK functionalities for image segmentation and registration and provide MATLAB integration.

SciRun includes several PowerApps which are software applications, developed using SciRun, but focused on specific application areas and/or tasks. A PowerApp allows building an interface based on an existing module network. While the dataflow network is responsible for the execution stage, the various interface windows of each module are grouped in a single interface window showing only the most important parameters. The other parameters are assigned default values. This allows hiding the complexity of the underlying network from end-users and the PowerApp can be run resembling a stand-alone software application.

One of the available PowerApps, BIOImage (figure A.9), allows the visualization of volume data (e.g., CT) providing 2D (axial, sagittal and coronal) and 3D views (volume rendering). It also allows re-sampling, cropping and median filtering.

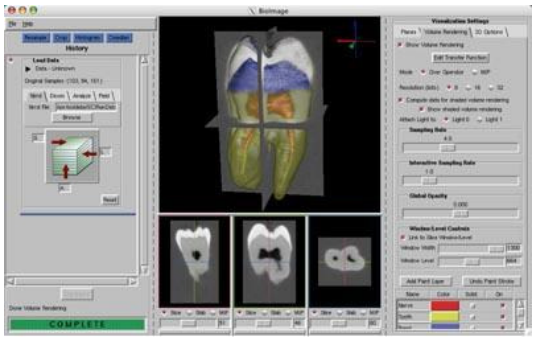


Figure A.9 – User interface of BioImage, a PowerApp provided by SciRun for 3D image sets visualization, resampling and croping.

Other applications developed at University of Utah which have evolved from SciRun's features are Seg3D (figure A.10) and ImageVis3D. Seg3D allows image processing and segmentation using

[23] METK – The Medical Exploration Toolkit: `www.metk.net/`

[24] SciRun – A Scientific Computing Problem Solving Environment: `www.sci.utah.edu/cibc/software/106-scirun.html`

algorithms from ITK. It is also possible to perform manual segmentations and image volumes can be labelled and viewed using volume rendering and axial, sagittal and coronal views. On the other hand, ImageVis3D is focused on visualization providing volume rendering of large data sets.

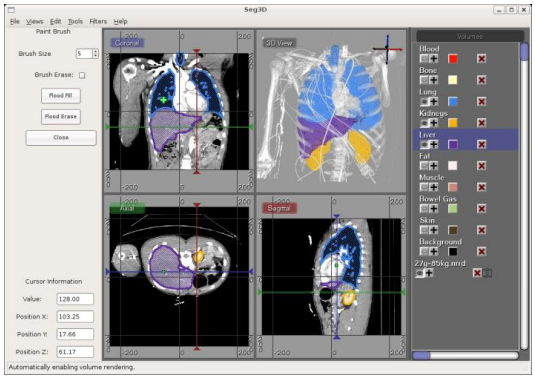


Figure A.10 – User interface of Seg3D, a software application which allows image processing and segmentation.

XIP – eXtensible Image Platform

XIP[25][161] is a software tool developed as part of the caBIG initiative[26] which is very similar to MeVisLab. It allows defining an image processing/visualization pipeline using a graphical interface by connecting several modules. These modules include DICOM reading and writing, volume rendering and ITK and VTK functionalities allowing fast prototyping without writing a line of code. Figure A.11 presents XIP's user interface showing a module network for smoothing an image and presenting it side-by-side with the result. Image data is read (SoItkImageFileReader), its intensity is rescaled (SoItkRescaleIntensityImageFilter), then it is smoothed (SoItkDiscreteGaussianImageFilter) and the remaining modules concern visualization.

Each module has a configuration panel which allows accessing its inputs and outputs along with a set of customizable parameters.

XIP also allows building modules to encapsulate complex networks allowing their reuse and contributing for better understanding wider networks. In the module network presented in figure A.11, the Package module is, in fact, an encapsulation of a more complex network. New algorithms can be tested by adding user built extension libraries, written in C++.

[25] XIP – eXtensible Imaging Platform: `www.openxip.org`
[26] `https://cabig.nci.nih.gov/`

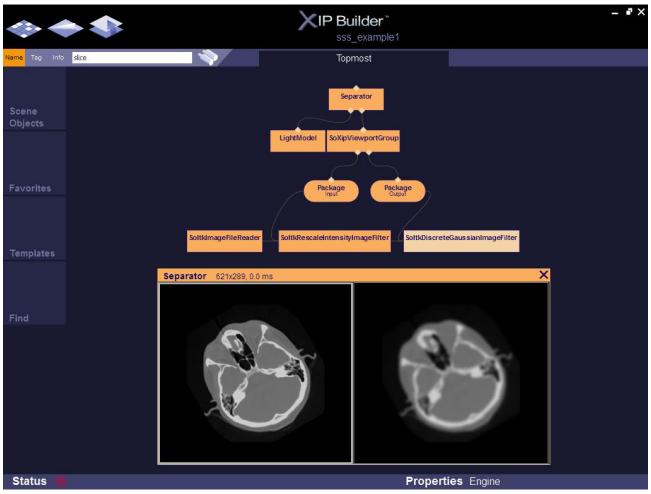


Figure A.11 – XIP's user interface showing a network for smoothing an image. The smoothed image is then shown side-by-side with the original.

DeVIDE

The Delft Visualisation and Image processing Development Environment (DeVIDE[27]) [162] is an application builder developed using C++, for the processor intensive tasks, and Python for the high-level features. It provides several visualization and image processing features by integrating several libraries, such as VTK and ITK, and providing higher-level functions to ease their usage. A very interesting feature of DeVIDE is that it can be used in two distinct modes: interactive and command line. In interactive mode (see figure A.12), DeVIDE is just like any application builder. It is possible to graphically build a processing pipeline by adding different modules and establishing connections between them while debugging the results and including the necessary adjustments. After building such processing networks it is then possible to run them from the command line, without the GUI. This way, the developed processing networks can even be used as modules in other tools.

New features can be added to DeVIDE through a multitude of options which include new modules (to be included in the pipeline), adding code to a module provided with DeVIDE called CodeRunner or by adding code snippets to existing modules (using the so called introspection interfaces) in order to modify their behaviour.

[27] DeVIDE: `graphics.tudelft.nl/Projects/DeVIDE`

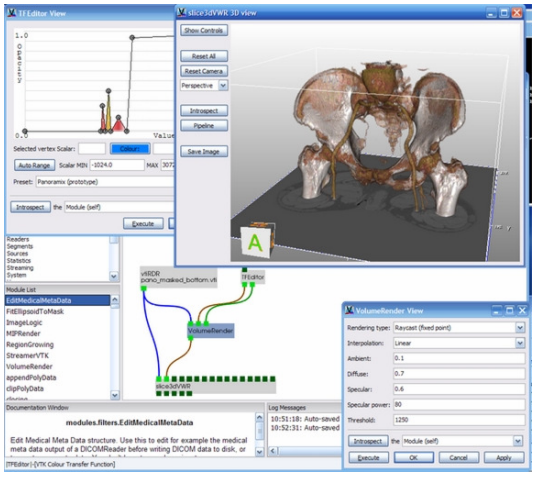


Figure A.12 – DeVIDE user interface showing a small processing network and a 3D visualization window.

A.4 Discussion

This section performs a comparison between the different libraries and software tools presented above, analysing some of their advantages and disadvantages.

To support the discussion a set of tables gathering additional information concerning the analysed tools are provided.

Table A.1 presents a list of some features we considered important to be natively supported by medical image processing tools regarding six main feature groups: DICOM image reading and writing with support for multiple timepoints; image visualization using standard orthogonal views (sagittal, coronal, axial) and multiplanar reformat (MPR); image analysis including distance measures and synchronized side by side image viewing for comparison; 3D visualization using the most common methods, namely volume rendering, maximum intensity projection (MIP) and shading; simple noise removal; basic image segmentation including saving the resulting segmentations, and image registration. When supported by a tool a feature is signaled with the symbol $\star$.

Table A.2 presents a summary of the main features of each software tool regarding a more advanced usage. It covers several aspects we found relevant when choosing such a tool, including if it is open-source; if it is extendable, here considered only if the extensions can be developed aside from the tool and then inserted (as a plugin) as opposed to modifying the source code; the programming language used for the extensions or (if it does not support extension) source code; if it allows building a graphical user interface which provides the user with only the desired functionality, hiding the underlying complexity; if it provides user/developer oriented documentation.

Finally, Table A.3 provides information regarding notable application areas (with a note on

Table A.1 – Summary of native features, deemed important, supported by the analysed integrated environment tools.

Operations		3D Slicer	AMIDE	BioImage	MIPAV	MedINRIA	ImageJ
DICOM	Read	⋆	⋆	⋆	⋆	⋆	⋆
	Write	⋆	⋆			⋆	⋆
	Volumes	⋆	⋆	⋆	⋆	⋆	⋆
	Time Series				⋆		⋆
2D Vis.	Ortho. Views	⋆	⋆	⋆	⋆	⋆	⋆
	Custom W/L	⋆	⋆	⋆	⋆	⋆	⋆
	MPR	⋆		⋆	⋆		⋆
3D Vis.	Vol. Render.	⋆	⋆	⋆	⋆	⋆	⋆
	MIP			⋆	⋆	⋆	
	Shaded			⋆	⋆	⋆	⋆
	Custom LUT	⋆	⋆	⋆	⋆	⋆	
Analysis	Dist. Measure						⋆
	Synch. View			⋆			
Noise Rem.	Smoothing	⋆	⋆	⋆	⋆		⋆
Segmentation	Thresholding	⋆	⋆	⋆	⋆	⋆[1]	⋆
	Reg. Growing	⋆		⋆	–	⋆[1]	⋆
	Volume Assess.			⋆			
	Saving	⋆	⋆	⋆	⋆		⋆
Registration	Img. Align.	⋆	⋆	⋆	⋆	⋆	⋆

⋆- Natively supported feature

[1] - Applied to specific tasks concerning brain image processing

some of the image modalities used) of each tool and a small set of recent literature references concerning works that use them. These references are not to be considered completely representative of the works carried out with each tool, but aim to provide pointers for research groups working with them and common tasks accomplished.

A.4.1 Toolkits

ITK is a very powerful toolkit for image processing, segmentation and registration. It is widely used and it might only be limited by the absence of visualization options. VTK can be used for this purpose but has limitations concerning complex interactions and interfaces. Both MITKs have been developed to gather, in a single toolkit, image processing methods along with visualization and interface design options. A big disadvantage of the MITK proposed by Tian et al. [148] is that it does not integrate ITK and VTK but their own set of image processing and visualization methods. So, Wolf et al.'s MITK seems a good toolkit for application development integrating

Table A.2 – Summary of the main features of several free/open-source tools for medical image processing, analysis and visualization.

Tool	Open Source	Extendable	Lang.	GUI (end user)	Doc.
Integrated Environments					
3D Slicer	YES	YES Cmd. line programs added as plugins	C++ Tcl	NO	YES User/Dev. (incomp.)
AMIDE	YES	NO	C++	NO	YES User
BioImage	YES	NO	C++ Tcl	NO	YES User
MIPAV	YES	YES Plugins	Java	NO	YES User
MedINRIA	NO	NO	–	NO	YES User
ImageJ	YES	YES Plugins	Java	+/-	YES User/Dev
Application Builders					
MeVisLab	+/-	YES New modules	C++ Java	YES	YES User/Dev.
SciRun	YES	YES New modules	C++	YES PowerApps	YES User
XIP	YES	YES New modules	C++	not clear	YES User/Dev. (incomplete)
DeVIDE	YES	YES New Modules	C++ Python	NO	YES User

ITK and VTK. When considering image-guided surgery scenarios, IGSTK can be a good choice. MARVIN seems to be evolving well and can be a good alternative in the future.

With some additional effort the developer might also use KWWidgets or FLTK to develop a user interface integrating VTK and ITK with more customized functionality.

A.4.2 Integrated Environments

All the analysed integrated environments have a considerable number of native features (Table A.1).

3D Slicer seems, from the compared tools, the one that offers more stability. It has already been used in a considerable number of works in a wide range of applications as can be observed

Table A.3 – Notable applications and recent literature references of works using the analysed tools.

Tool	Notable Applications	References
3D Slicer	Analysis/Seg./Registration Virtual Endoscopy / 3D Vis. Surgical Planning / Assistance CT/MRI	Lodygensky et al. [163] Nakajima et al. [164] Raappana et al. [165]
AMIDE	Analysis / Registration Singular Value Decomp. SPECT/CT/PET/MRI	Israel-Jost et al. [166] Naum et al. [167] Yushmanov et al. [168]
BioImage	Analysis/Segmentation Registration Smoothing / Resclicing MRI/fMRI/DTI	Zhu et al. [169] Laufer et al. [170] Whang et al. [171]
MIPAV	Segmentation Volume assessment User developed features MRI/PET/CT	Bazin et al. [172] Calabrese et al. [173] Faustini et al. [174]
MedINRIA	Segmentation/Registration Brain Images Analysis Diffusion Tensor Imaging MRI/DTMRI	Choi et al. [175] Yassa et al. [176] Peyrat et al. [177]
ImageJ	Analysis/Segmentation Image conversion Diffusion Tensor Imaging MRI/DTMRI/CT	Barboriak et al. [178] Burger et al. [179] Koompairojn et al. [180]
MeVisLab	Visualization / Segmentation User developed algorithms Tools (e.g., MeVisPulmo) Multi-slice CT	Kuhnigk et al. [181] Preim et al. [182] Oeltze et al. [183]
SciRun	Vis. / Segmentation User developed features CT/MRI	Dedual et al. [184]
XIP	—	—
DeVIDE	Segmentation 3D Reconstruction Integration with other tools	Maheshwari et al. [185]

in Table A.3 and, with further detail, in its homepage.

AMIDE has been successfully used in several works to perform data set alignment. Nevertheless, its user interface could be improved and, as far as it can be seen, it is not under active development.

MIPAV still seems a bit unstable in some aspects and memory consumption is rather high which might be a problem when working with large data sets. Nevertheless, it is a user-friendly tool which works fairly well with small datasets and provides an interesting set of tools. It has been used with success in several research works (see Table A.3). The fact that it is written in Java might also be an advantage for some users, more used to this programming language, but some performance issues seem to arise. Another good option for Java programmers is ImageJ. It is a robust and lightweight tool with a large number of native features and with a large community constantly developing new plugins. The fact that it allows recording macros (i.e., sequences of operations that can later be automatically repeated) and batch processing (automatically process a set of images) is an interesting advantage. ImageJ is, from the analysed integrated environments, the only one which allows something close to a GUI which hides complexity from users. Using the macro features it is possible to run the macros on startup or record macros and then present them as an option on the menu bar. Customisable tool sets can also be presented to the user (in a toolbar).

The BioImage Suite is also a powerful tool with a large set of native features but would benefit from a refreshed look and feel of the user interface. As far as can be observed in the literature (and given its existence time) it has not been used in many applications although extensive use at Yale is referred and the discussion forum is reasonably active.

MedINRIA is an interesting tool, with a good user interface, but still lacks a broader range of image processing features to be applied in general image processing scenarios (although it supports several tools for brain image analysis). The fact that it is the only tool which does not provide the source code (see Table A.2) or extension possibilities is also a disadvantage. If it continues evolving and integrating new modules it can become a tool of choice.

A.4.3 Application Builders

Regarding the application builders, SCIRun is a generic computation and visualization tool. Even though it provides some ITK integration, given its design focused on versatility, it might not be the first option data flow tool for medical imaging and its PowerApp, BIOImage, would benefit if it was extended with a wider range of processing tools. Seg3D can be a choice as a segmentation tool.

Comparing XIP with MeVisLab, it must be noted that MeVisLab has some advantages on

usability. In both tools each module has a configuration panel where several parameters can be adjusted / viewed. When the user wants to connect two modules (e.g., SoItkImageReader and SoItkRescaleImageFilter) it is necessary to open the panels for both modules and then drag the output field of the first over the input field of the latter. In MeVisLab this is easier because the module has the inputs (bottom of module) and outputs (top of module) drawn on it and there is no need to open the configuration panels. This also contributes to a better understanding of the data flow.

One feature provided by XIP which might get handy is the possibility of selecting a set of modules in a network and pack them inside a single module (which can be unpacked at any time) to "unclutter" the network. MeVisLab supports macro modules but additional work is required to define its inputs and outputs formally.

DeVIDE has a look and feel which is more intuitive than XIP. The inputs and outputs are depicted graphically on the modules which helps to visualize the data flow. The modules are stored in a more organized fashion, by theme (e.g., readers, filters, viewers, etc.) and it is easy to understand which kind of data should go on each module input. A disadvantage, when compared with XIP and MeVisLab, is that it is not possible to connect parameters between modules.

At this moment, MeVisLab provides a wider range of integrated ITK methods and a wider set of macro modules providing a set of features which ease image data visualization such as 2D/3D viewers or orthogonal planes viewers. In fact, almost all items presented in Table A.1 are supported in MeVislab by just adding one module to a network which is not the case with the other tools. This advantage is clearer when comparing with XIP where building a visualization network for an image set, for example, is not very intuitive. Including some visualization modules which hide some of the complexity would improve XIP. Regarding visualization, DeVIDE allows fast visualization but still lacks more options, (e.g., 2D orthogonal planes).

A.5 Conclusion

This appendix presents an overview on several free and open source tools for medical image processing, analysis and visualization.

Some of these tools described in the literature have been left out of this analysis because they are not publicly available or the available version is still very raw, not providing the full extent of the announced features (e.g., CAVASS [186] [187]).

ITK is clearly a very good source for image segmentation and registration methods. It is widely used and very good documentation is provided which includes not only a reference to the available filters but also many application examples. Given the amount of auxiliary tools available

(such as MeVisLab), first experimentation with ITK or simple usage of the different features can be performed without a need for programming skills.

For building end-user applications, MITK by Wolf et al. [147] is a serious option followed by KWWidgets when there is a need for developing specifically tailored new interfaces. Nevertheless, given the support for Qt widgets provided with the MITK toolkit this will probably be seldom needed.

For analysis of an image data set and some non repetitive (applying a large number of operations to several data sets might be a tiresome task) image processing, 3D Slicer or Yale's BioImage Suite seem the most appropriate tools with the latter having a "less friendly" user interface and not supporting a plugin based extension which, in general, would ease the task of adding new features. If the goal is to develop new features, the tools supporting extension without having to change the source code have an advantage.

ImageJ is adequate for batch processing given its macro support. Although it does not provide such a convenient environment for medical image processing (given its more versatile nature) it exhibits better performance and response times than MIPAV. Furthermore, several plugins exist for a large set of purposes.

For developers or even users with no programming skill, MeVisLab is a very good option for fast prototyping as it integrates a huge range of ITK filters (from which users acquainted with ITK can benefit) and several visualization and processing macros. Extending MeVisLab is also easy and allows, for example, reusing previously written code using ITK although a conversion between ITK and MeVisLab image formats is required but easily accomplished. The good documentation for MeVisLab and ITK is also a great advantage. MeVisLab also has the advantage of allowing the development of an user interface to more easily interact (hiding the complexity) with the created processing network.

Concerning user-developed features integration, MeVisLab and MIPAV seem the most appropriate (judging by some of the works presented in the literature) and MIPAV (closely followed by ImageJ), as it is developed in Java, might be a good choice for those who are more proficient in that programming language. DeVIDE, with the possibility of running a created processing network from the command-line is a good option for developing and testing new features in a visual environment and then integrate them as "black-boxes" in third-party applications.

An interesting set of tools which can be used in a development pipeline might be ITK for image processing, MeVisLab for initial prototyping and additional module development and Wolf's MITK for building an end-user application. By using ITK to create additional modules in MeVisLab they can be ported with minimal effort from MeVisLab to MITK.

This is a crucial aspect. A recurring problem when trying to go from research to clinical

applications is that the existing prototype has been developed in a platform that is not suitable to be used in a clinical scenario, e.g., due to performance issues or the lack of proper user interaction features. This often requires a full reimplementation of the proposed methods using a different framework and, given the costs involved in such a task, the method might never have a chance to show its full potential.

We, therefore, argue that care should always be taken in order to choose the best tool for each development stage considering, for example, time schedules, project goals and developer's technical skills while considering how this choice can better serve future developments.

Finally, the analysed tools seem to have a considerably low learning time (at least to accomplish basic tasks and given the documentation provided) and there is also a large community using these tools (most of them under strong active development), willing to help and share their experience and providing constant validation.

Appendix B

CardioAnalyzer

"Vai agora com um machado de bronze cortar grandes troncos
para fazeres uma ampla jangada. Sobre ela fixa uma plataforma,
a parte mais elevada do casco que te levará sobre o mar brumoso."
Homero, Odisseia, Discurso de Calipso a Ulisses

THE effort involved in developing the different features presented in the last chapters depended on how much of the previous work could be reused at each stage. In addition, one of the main options during development was to support the discussion with the radiographers and clinicians by presenting them working prototypes. This required a framework which could be continuously adapted to tackle different needs such as changes in the graphical user interface to support new features or access to segmentation data.

To account for the different needs explained above, a software application based on the MITK, ITK and Qt libraries has been developed, named CardioAnalyzer, to support development and testing of the different features concerning left ventricle (LV) segmentation and analysis.

The main purpose of this appendix is to present CardioAnalyzer. Its main focus concerns features not described in the previous chapters and different aspects of the graphical user interface. A brief account on how these can be used to perform the different tasks is also provided.

Publications:

The work presented in this chapter has been partially published in:

- Samuel Silva, Joaquim Madeira, Beatriz Sousa Santos, Augusto Silva, "CardioAnalyser: A Software Tool for Segmentation and Analysis of the Left Ventricle from 4D MDCT Images of the Heart", Proc. 7th International Conference BioMedical Visualization (MediVis10, pp. 629–634, London, United Kingdom, July, 2010

B.1 Introduction

Developing a software application to support research can be a tiresome task, since goals and requirements change along the way. Small prototypes can be developed from scratch (throw-away prototypes) to test different ideas, but this option is not always the best as it requires a lot of effort. This is particularly true when the end users must be able to use those prototypes and interaction techniques, along with processing methods, are also to be evaluated. This requires much more than just simple graphical user interface prototypes and, furthermore, it is desirable that the gap (in terms of time taken and development effort) between the prototype and a clinically usable application be small. Therefore, we adopted an evolutionary prototyping strategy [188] aiming to build a software application with a conceptual model able to broadly accommodate a set of general requirements and then use it to support development.

An evolutionary prototyping strategy is also relevant as it provides advantages concerning the time needed to deliver new features (vital in a time limited development scenario) and users involvement with the system [188]. The latter, resulting from their inclusion in the development process, also possible by the availability of usable (available features, performance, etc.) software tools, might also motivate users to feel a part of the whole process and to cooperate to see it work.

The work carried out included the development of different features for LV segmentation and analysis. After an initial prototyping stage, using MeVisLab[1], a new software application was developed. The main requirements for this new application concerned two main aspects:

- Provide a framework for developing different interaction techniques, include and manage multiple data from different analysis methods and control system performance by being able to work at different application levels from exam loading to visualization;

- Provide a software application which could be used as a platform to discuss ideas with radiographers and clinicians and which they could use for testing purposes.

Concerning the last requirement it is important to note that a discussion with the users, based on a working system, is also important to clarify any misunderstandings between them and the developer [188].

Therefore, while not intending to be a "finished tool" for routine clinical use it should provide an intuitive user interface and support users along tasks. It must be noted that in several occasions radiographers and clinicians used the application on their own (segmentation evaluation protocol, myocardial perfusion assessment) and, therefore, the application should provide easy, direct access

[1] MevisLab: `http://www.mevislab.de`

to the features being tested and allow blocking features not being evaluated without affecting overall application functionality (e.g., blocking analysis features during segmentation evaluation protocol).

Following on the main conclusions presented in appendix A, we chose the Medical Interaction Toolkit (MITK)[2] to provide the core features (data handling and image visualization) of our framework, Qt[3] for user-interface development, ITK[4] for image processing and OpenGL[5] for rendering. All development has been performed using C++.

In the following sections we present a more detailed description of the developed software application, CardioAnalyzer. The purpose is to explain the main ideas behind its architecture, give some details concerning features which were not described earlier and to show the graphical user interfaces in more detail.

This chapter is structured as follows. Section B.2 describes the main ideas behind the different aspects of the graphical user interface (GUI) presenting some low-fidelity mock-ups. Section B.3 overviews additional features of CardioAnalyzer, not described in the previous chapters, and presents the GUIs supporting CardioAnalyzer's different features and highlighting relevant details. Finally, section B.4 presents some conclusions.

B.2 User Interface Conceptual Model

Considering the main requirements presented in the last section our aim was also to provide a software application simple to use (for the end-users and developer), keeping coherence among the different aspects of the GUI, and providing a clear visibility of the main features while hiding those not relevant for current system status [189].

CardioAnalyzer's GUI is organized in four areas as depicted in the low fidelity mock-up presented in figure B.1. There are two main areas: a data and image viewing area where, for example, the image view planes will be located; and an area where the widgets to support the different features (tools) will be made available. The two remaining areas are assigned to menus/toolbars and to system status feedback (idle or busy).

Figure B.2 presents a low-fidelity mock-up for CardioAnalyzer's main screen. The main screen should show a list of available exams (patients) and allow access to the different tools provided. Since CardioAnalyzer is a software application to support continuous development of new features, a simple method should be devised to: 1) provide easy access (and integration) to new features (tools) and 2) ensure their visibility providing a clear entry point to using CardioAnalyzer.

[2]Medical Imaging Interaction Toolkit: `http://www.mitk.org`
[3]Qt library: `http://qt.nokia.com/`
[4]Insight Segmentation and Registration Toolkit: `http://www.itk.org`
[5]OpenGL library: `http://www.opengl.org`

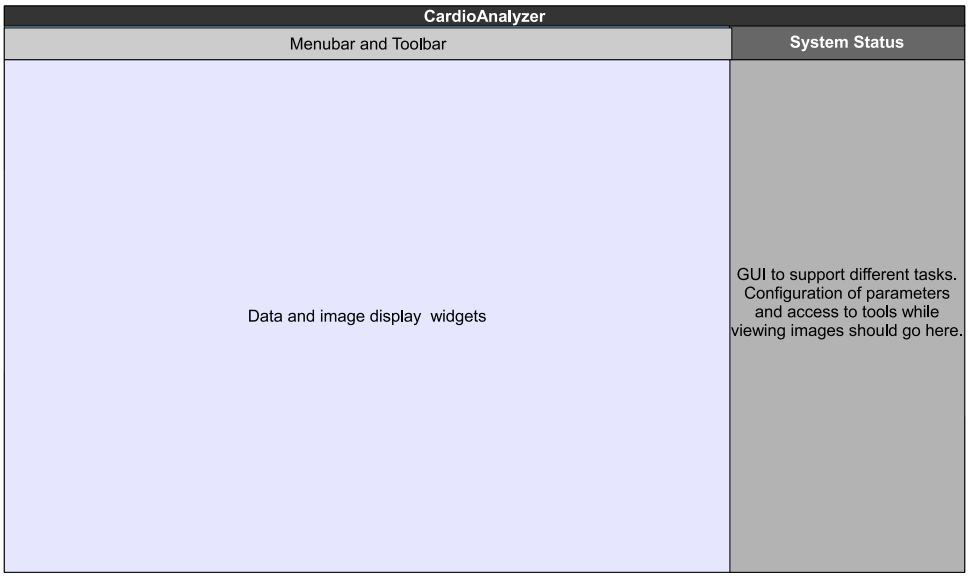


Figure B.1 – Low fidelity mock-up of the graphical user interface of CardioAnalyzer depicting screen space allocation.

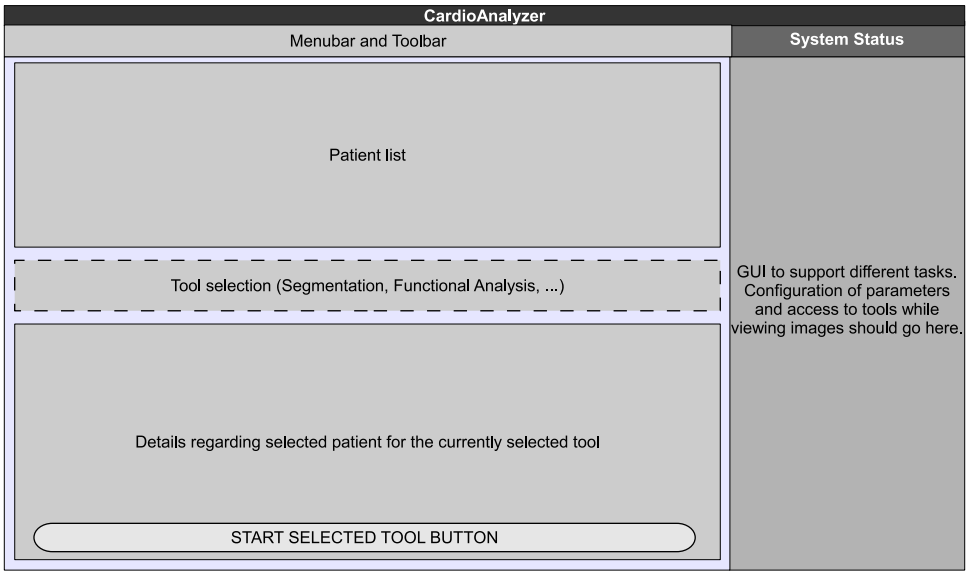


Figure B.2 – Low fidelity mock-up of the graphical user interface of CardioAnalyzer depicting the main screen.

Therefore, access to the different tools (segmentation, functional analysis, etc.) is provided in an area just below the patient list (e.g., a button for each tool). The bottom area provides additional detail concerning the currently selected patient and tool. For example, when a patient is chosen and the segmentation tool is selected, the details area might show which segmentations have already been performed.

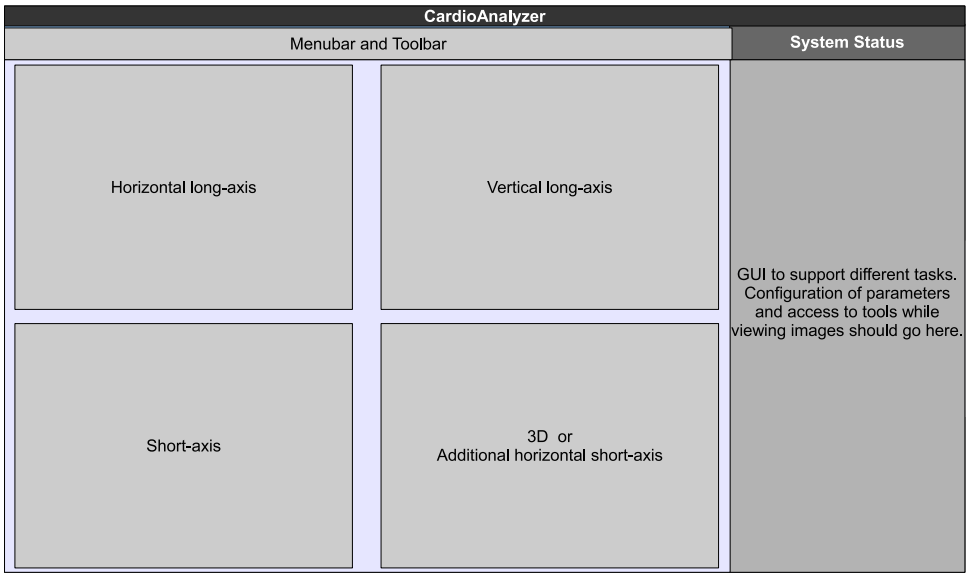


Figure B.3 – Low fidelity mock-up of the graphical user interface of CardioAnalyzer depicting the image visualization/inspection area.

Figure B.3 shows the mock-up for the image visualization/inspection area. Four view windows are provided one for each of the three standard cardiac analysis planes and a fourth for a 3D view or an additional horizontal long-axis view.

With the mock-up presented in figure B.2 the existence and access to multiple (and possibly increasing along development) tools is solved by a set of buttons positioned below the patient list. But the plan was also to have multiple (increasing during development) LV functional analysis parameters. To deal with this issue, figure B.4 presents two low-fidelity mock-ups for two different situations. In the first, the user is able to inspect the image and choose, from a list, the parameter to be displayed. Therefore, simultaneous visualization of the image and one analysis parameter is possible. In the second situation, the user is able to compare the representations of multiple parameters side-by-side.

In this way, the drop down list on the right provides access to any parameter available, with the representation of the corresponding data below, and the analysis board allows the user to compare any number of parameters side-by-side.

B.3 Main Features and Graphical User Interface

This section provides details regarding features provided by CardioAnalyzer, which were not described in the previous chapters, along with a brief description of the GUI supporting the different tools (segmentation, functional analysis and myocardial perfusion assessment).

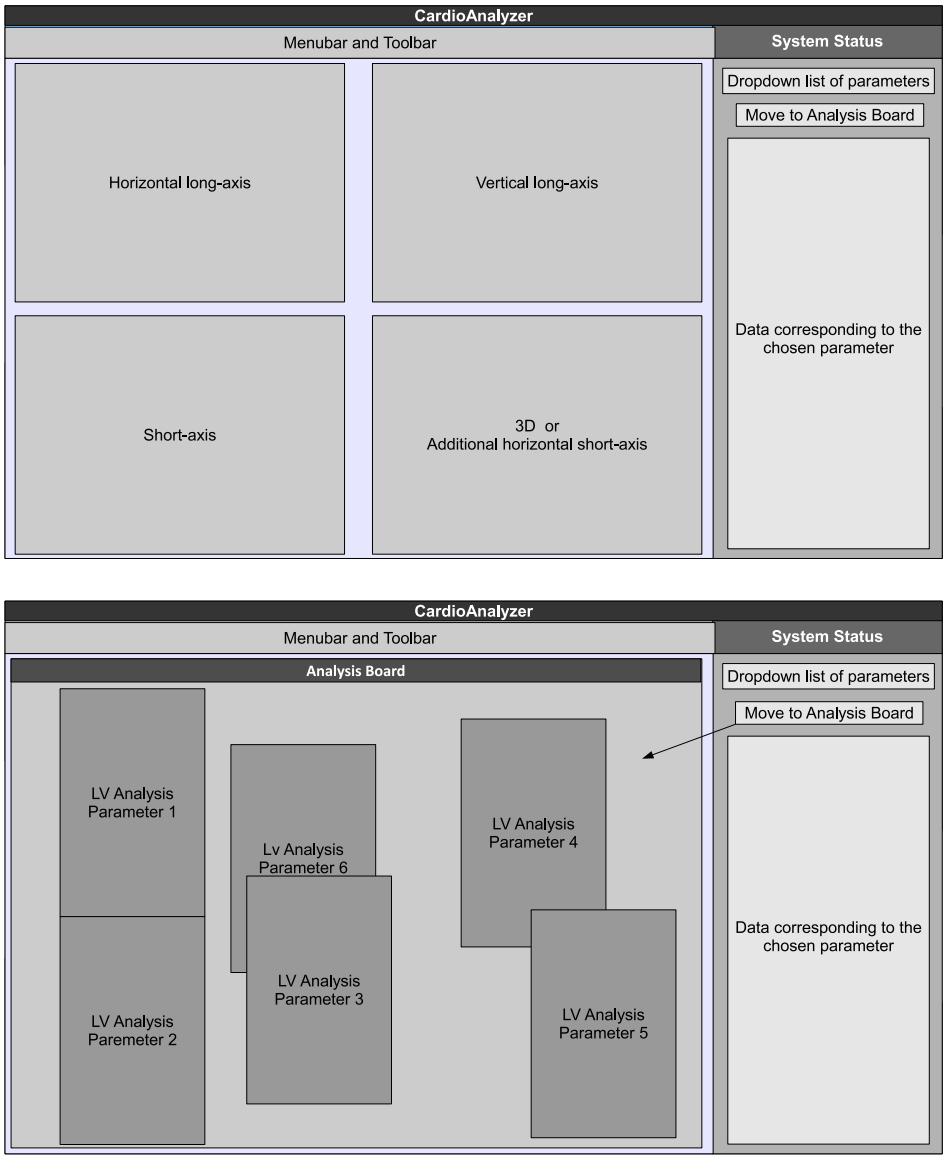
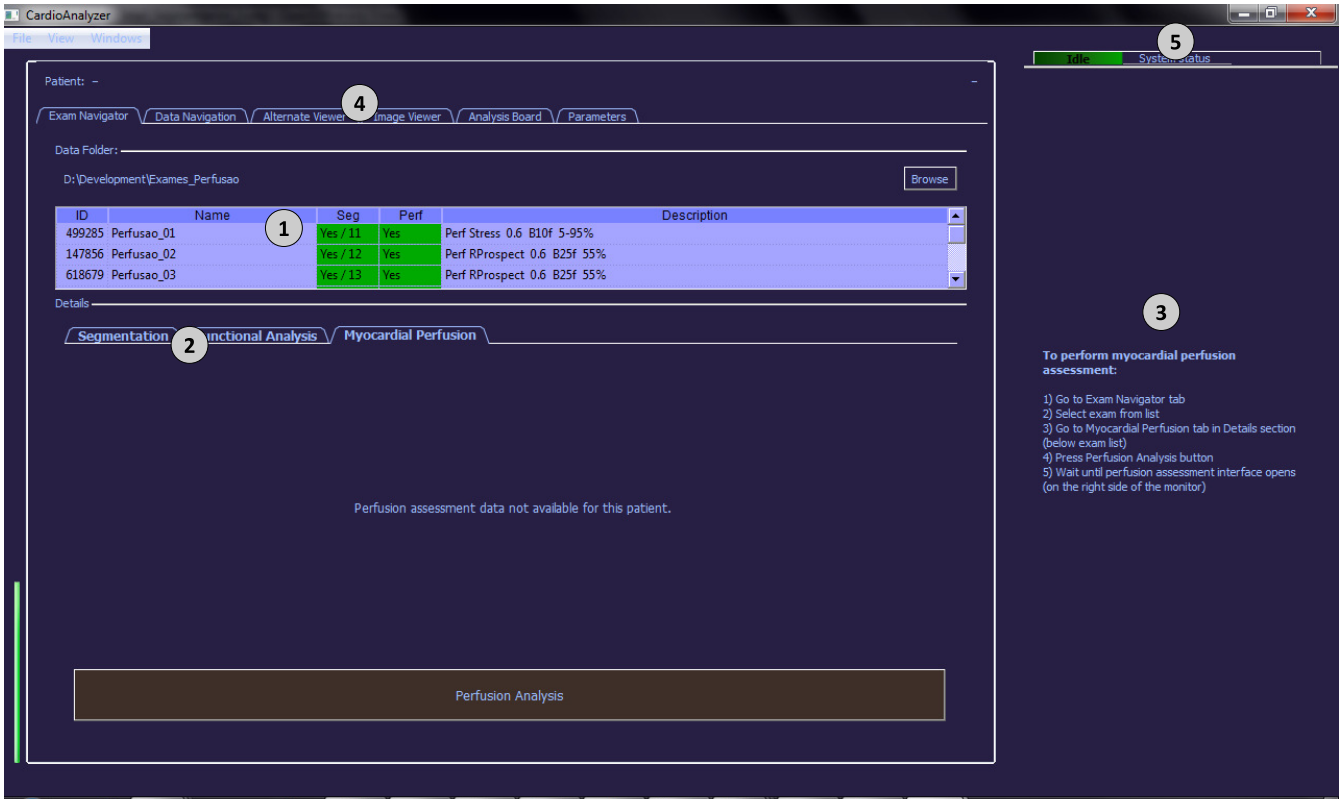


Figure B.4 – Low fidelity mock-up of the graphical user interface of CardioAnalyzer depicting two different scenarios to deal with LV analysis parameters.

B.3.1 Image Data Management and Processing

As previously mentioned, CardioAnalyzer is based on the MITK library. The most important feature provided by MITK is the data tree.

The data tree allows managing the different image volumes loaded (e.g., for different cardiac phases) the different segmentation and selection masks, point sets or even surface meshes. This can be done hierarchically, for example "hanging" the segmentations on the corresponding image volume. The main result is that when one of the images is selected for visualization its children



1. Patient list
2. Tabs to access segmentation, functional analysis and myocardial perfusion assessment features
3. "Get started" instructions
4. Tabs to access image viewer and analysis board
5. System status indication: idle or busy

Figure B.5 – Main screen of CardioAnalyzer when the application is started.

will also be made visible. For each element in the data tree several properties can be changed such as colour, opacity or if it is an image to be represented as binary. Furthermore, MITK provides a set of classes which allow loading DICOM images and automatically place them on the data tree.

All the image processing methods have been developed using ITK. Since MITK is based on ITK, no image format conversion was needed (just type casts) and, therefore, no major overheads existed when going from one format to the other, for example to process an image stored in the data tree (MITK image format) using ITK.

B.3.2 Main Screen

The access to the different tools, instead of being performed using a button for each of them, as initially proposed in the low-fidelity mock-up, is implemented using a tab widget, i.e., a set of stacked widgets (one for each tool) with a corresponding tab (figure B.5, detail (2)). When a tab is clicked, the corresponding tool is selected and the GUI showing relevant details is shown. This option makes it easier to add additional tools and manage GUI updates.

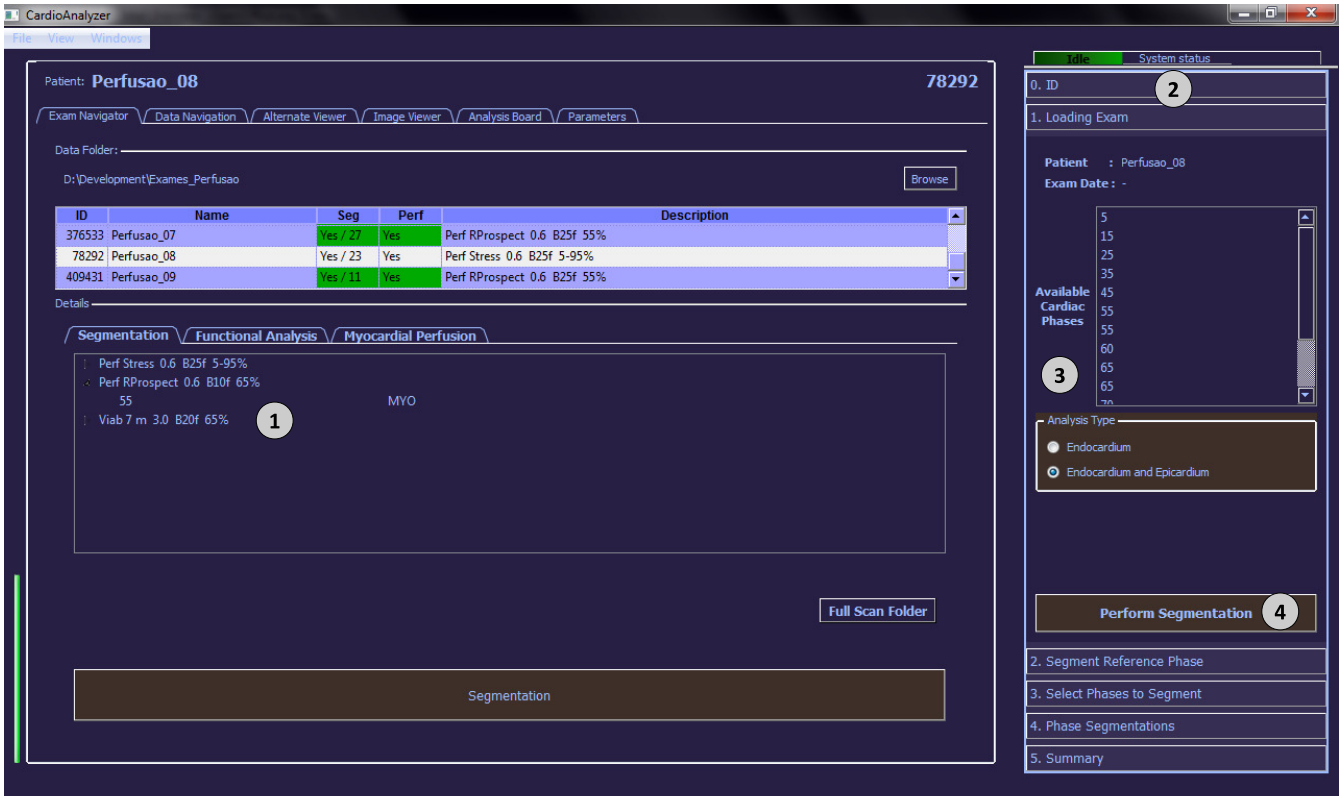
When none of the main "tools" (segmentation, functional analysis or myocardial perfusion assessment) has been started, the right side of the screen (where the GUI presents the different elements supporting the available features for each tool) presents a set of instructions on how to get started, i.e., the steps to follow in order to use the selected tool. In figure B.5, detail (3) guides the user on how to use the myocardial perfusion assessment tool.

Finally, when CardioAnalyzer is performing some time consuming task (e.g., reading files from disk, computing functional analysis parameters) a "status flag" showing "Busy" is presented on the upper right corner (figure B.5, detail (5)) providing the user with feedback on system status [190]. This feedback is also provided, in many situations (in particular those features in a more mature state), using a dialog box with a short message concerning the task being performed and a progress bar (e.g., loading exam). Nevertheless, adding a dialog box and progress bar to provide feedback requires some development effort and the status flag is a simple and easy method to do it.

B.3.3 Exam Browsing and Loading

Instead of requiring the user to select a folder where the DICOM[6] files for an exam are located, in order to open it, we considered important to provide a list of patients available for analysis (extracted from all the exams present in a folder which worked as the exam repository) as this is easier to manage (even during development). This list includes the patient name and a brief

[6]DICOM – Digital Imaging and Communications in Medicine: further information regarding the DICOM format can be obtained at `http://medical.nema.org/`.



1. List of studies contained in the exam and performed segmentations
2. Several numbered tabs concerning the different stages of the segmentation
3. List of cardiac phases included in the exam and choice of endocardium only or endocardium + epicardium segmentation
4. Start the segmentation.

Figure B.6 – Graphical user interface after selecting the segmentation tool tab and choosing a patient from the list.

description of the exam type (coronary angiography, stress, etc.). Figure B.5 shows (detail (1)) the list of patients. This list also provides information regarding the existence of segmentations and myocardial perfusion assessment data for each patient.

Loading an exam stored in DICOM format includes two main stages: scan the DICOM files that compose it in order to read different DICOM tags[7] (e.g., patient name and series identification) and then read the image content. Reading image content can be performed selectively, choosing a subset of the image data to load, e.g., a few selected cardiac phases instead of the whole exam. The main issue with the DICOM format is that a coronary CT angiography (CTA) exam might contain around 3000 files since each axial slice is stored in a separate file. Therefore, the initial scanning stage implies opening 3000 files, which takes a considerable time. Furthermore, opening each cardiac phase includes reading around 300 files. For developing purposes, an exam might need to be opened dozens of times, having a clear impact on development time, and a long loading time (much higher than the typical loading time in dedicated workstations) might also have a negative impact on user satisfaction.

To tackle this issue, the first time any DICOM exam is read an index file is created including a list of the cardiac phases (and corresponding relevant DICOM tags). Furthermore, when each cardiac phase is read, the resulting image volume is stored to an NRRD[8] file and the corresponding DICOM files are deleted. The result is that when the exam is read a second time, instead of having to scan 3000 files for the DICOM tags, it just has to read the index file and loading each cardiac phase means opening a single NRRD file. This allowed a clear reduction in exam loading time from around six minutes to one minute in a 2.20 GHz dual-core CPU computer with a 5400rpm hard drive. This conversion can also be performed offline. For example, for the evaluation of the myocardial perfusion assessment (described in chapter 10), 20 exams were used and batch converted before the evaluation to provide the clinician with the fastest possible performance.

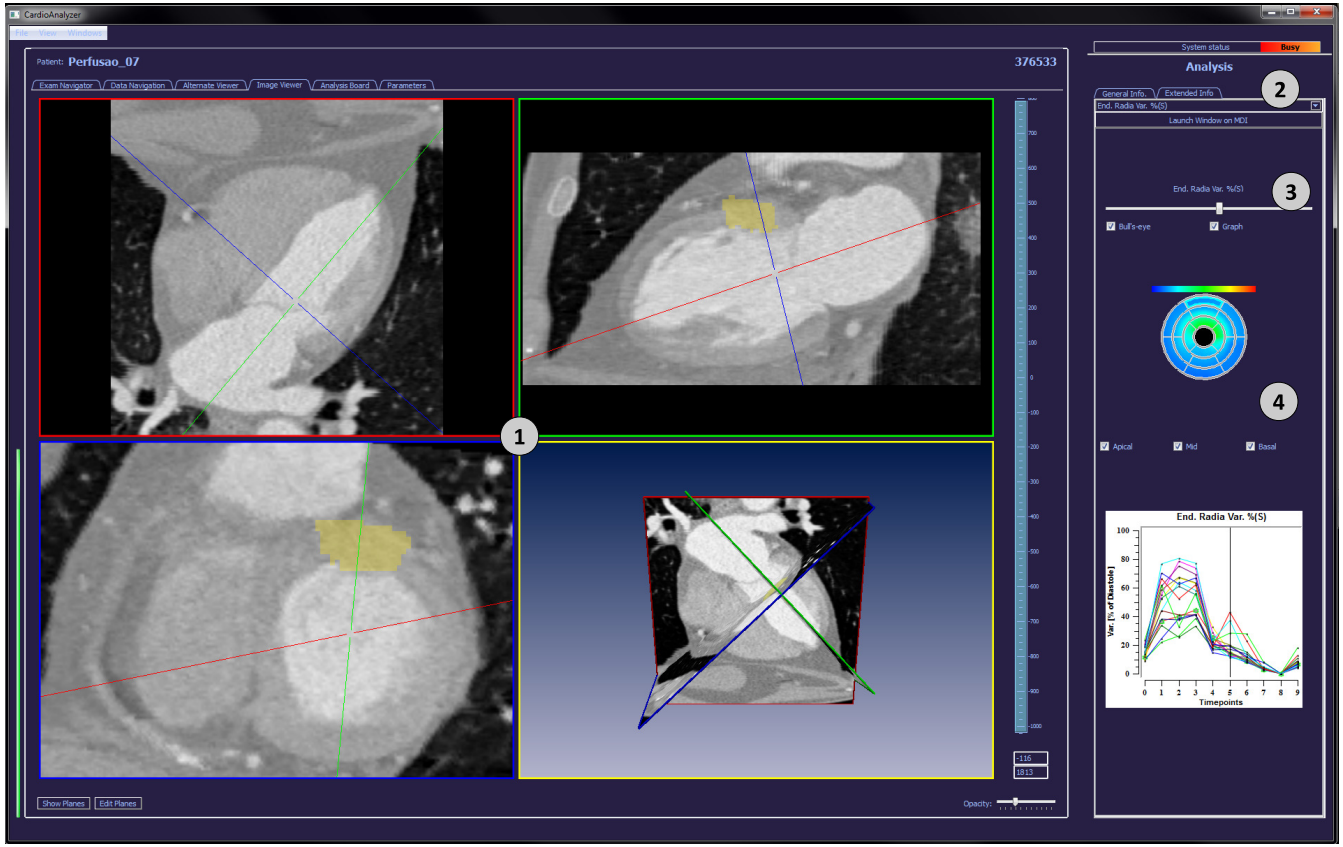
B.3.4 Image View Planes

The image view planes (or slicing planes, as they slice the image volume) are based on a Qt widget provided with MITK, allowing interactive definition of plane position, orientation and zoom. Moving any of the planes up and down along their normal vector direction is possible using the mouse wheel. This feature is often used for a quick inspection, for example, along the LV long axis. Using the mouse wheel for this purpose might require a considerable amount of user effort.

[7] Reading DICOM tags is not only important to identify the patient and exam type but also to allow the identification of the different cardiac volumes included. For example, the images belonging to each cardiac phase have a common value for tag `0018:0022` which corresponds to scan options. This allows identifying which DICOM files concern each cardiac phase.

[8] NRRD image format: Nearly Raw Raster Data. Detailed description available at `http://teem.sourceforge.net/nrrd/`.

Figure B.7 – Graphical user interfaces supporting the different stages of the segmentation protocol.



1. Image view planes with myocardial segment highlighted
2. Drop down list of computed LV analysis parameters
3. Slider to move along the different cardiac phases
4. Representation of data obtained with the chosen parameter

Figure B.8 – Left ventricle functional analysis performed while viewing the image and the data for endocardium radii variation.

In order to provide an alternative we added support for plane movement by using the keyboard: by pressing and holding keys **1** or **2** the current plane (defined by where the user clicked the last time) can be moved up and down.

B.3.5 Left Ventricle Segmentation

Left ventricle segmentation features are accessible by choosing the proper tab in the details section. When a patient is selected a list of the studies available is presented (e.g., stress, rest, viability). Each of the studies can be expanded to show the cardiac phases included. For each of the cardiac phases information on the availability of a segmentation (performed earlier) is provided.

Figure B.6 shows the first stage of the segmentation feature. On the right side the user is provided with a list of the cardiac phases included in the exam and has to choose segmenting both endocardium and epicardium (default) or just the endocardium. This latter option is faster to perform and can already provide data for functional analysis (refer to chapter 9, e.g., global blood volumes, regional blood volumes, endocardium radius).

Figure B.7 shows details for different stages of the segmentation protocol. Detail (1) depicts the GUI for the segmentation of the reference cardiac phase. This stage includes cardiac phase selection, view planes definition and approval, threshold level tweaking, mitral valve plane adjustment, endocardium segmentation revision and approval, epicardium scale and threshold adjustment and epicardium revision and approval. For each step of the segmentation protocol, only the part of the GUI related with the current step to be performed is enabled (also exhibiting a brown background). The transition between stages is automatically performed after the last step of each stage is completed. Detail (2) concerns the selection of the cardiac phases to be segmented (all selected by default). Detail (3) shows the GUI for the segmentation of each cardiac phase. Apart from the steps for endocardium and epicardium segmentation revision and approval, this stage also presents a list of the cardiac phases selected for segmentation and a report of what has already been completed. A progress bar is shown, at the bottom, depicting the overall progress. Finally, detail (4) shows the final stage of the segmentation protocol presenting a summary of the segmentations performed and the corresponding blood volume for each phase.

An important feature to note is that all the performed segmentations are automatically stored on the hard drive. If for any of the cardiac phases to be segmented a previous segmentation (endocardium or epicardium) exists, CardioAnalyzer allows choosing between loading the existing segmentation or redoing the segmentation. In any of these cases, the user approved segmentation will overwrite any stored segmentation.

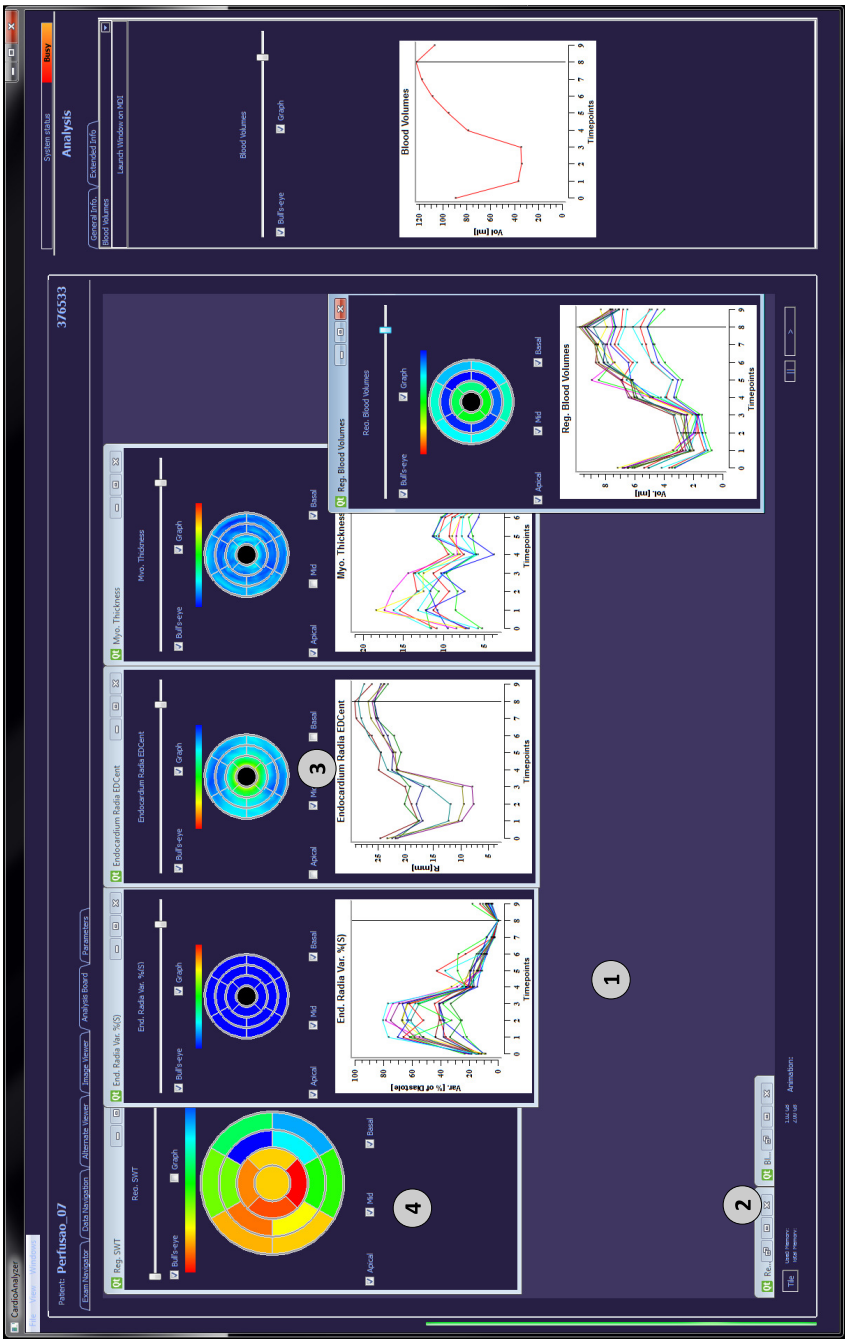


Figure B.9 – Analysis board allowing the simultaneous analysis of data for different parameters characterizing LV function.

B.3.6 Left Ventricle Functional Analysis

To perform LV functional analysis existing segmentations are required. When the functional analysis tool is started a list of the cardiac phases which have already been segmented is presented. The user then chooses those to be included in the analysis.

Figure B.8 shows the GUI for LV functional analysis, after the different parameters have been computed. The image view planes are shown (detail (1)) and the parameter currently being presented (detail (4)) can be selected using the drop down list in (2). The presented slider ((3)) can be used to navigate along the different cardiac phases. As previously mentioned, selecting any segment in the polar maps highlights the corresponding anatomical region in the image.

Analysis Board

If the user wishes to compare different analysis parameters side-by-side, the analysis board can be used. The analysis board is a multiple document interface (MDI) where windows depicting different analysis parameters can be viewed simultaneously. To add a window depicting one of the available parameters to the analysis board, the user can use the button shown in figure B.8, near detail (2), Launch Window on MDI. This button allows to create a window inside the analysis board for the current parameter being viewed.

Windows inside the analysis board can be manipulated in order to choose their position and size, or to remove them. As previously mentioned all these windows keep synchronized with each other, showing data for the same cardiac phase. Figure B.9, detail (1) shows the analysis board containing different windows. Notice that the different parameters are synchronized regarding the current cardiac phase but allow different configurations of what is shown. For example, detail (3) shows a line graph depicting only the curves for the mid-ventricular segments and detail (4) shows a polar map without the corresponding line graph. Detail (2) shows two minimized windows. Minimizing windows might help better organize the available space, temporarily removing some parameters from view.

B.3.7 Semi-Automatic Myocardial Perfusion

When the tool tab for myocardial perfusion is active, and a patient is selected, a summary of the myocardial perfusion assessment for that patient, if it has already been performed, is shown. Figure B.10 shows perfusion assessment summary for a patient. The two polar maps presented refer to the rest (on the left) and stress (on the right) studies. This is possible because the data concerning myocardial perfusion assessment is stored on disk. As previously mentioned, if the

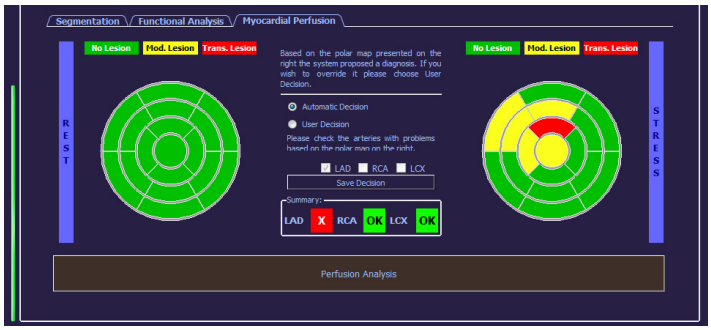


Figure B.10 – Details regarding a previously performed myocardial perfusion assessment showing the polar maps for rest (left) and stress (right) and diagnosis (middle).

user chooses to repeat the myocardial perfusion assessment any existing analysis will be discarded and the new analysis has to be performed from scratch.

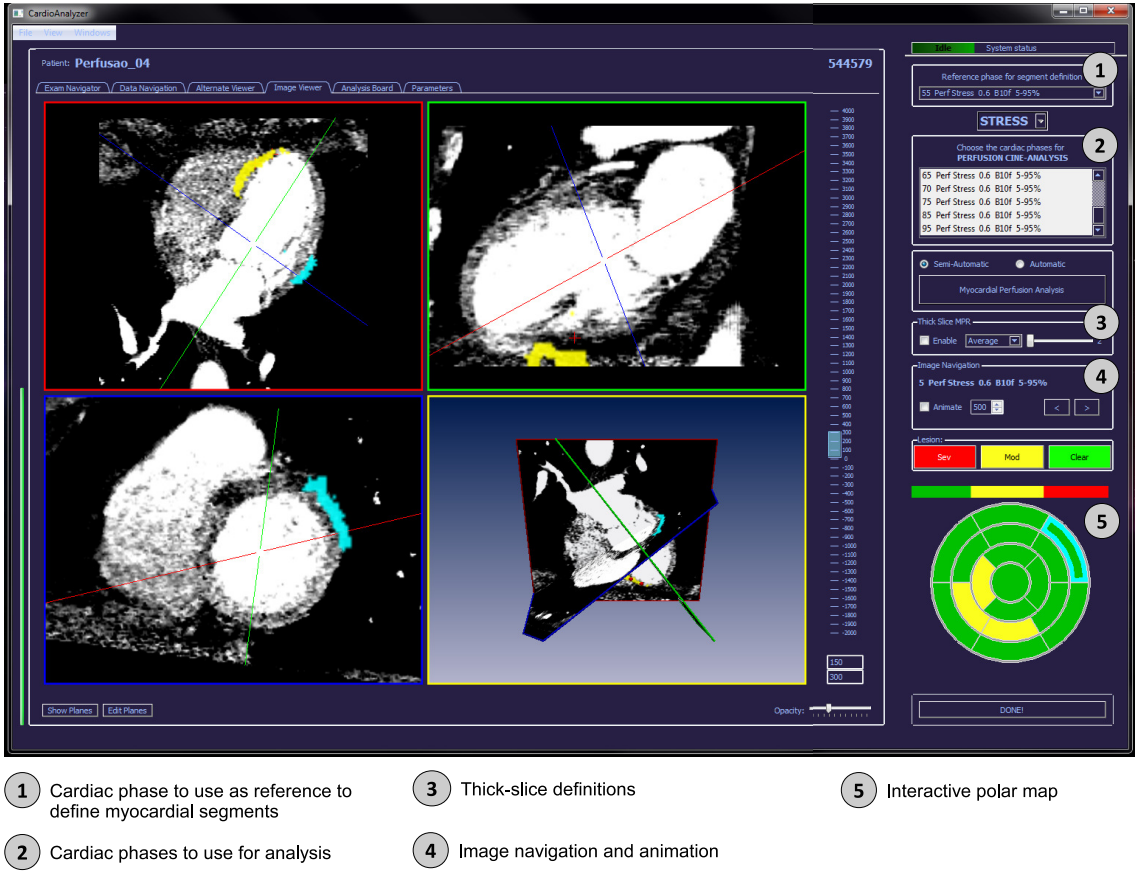
When the myocardial perfusion assessment tool is started the corresponding GUI appears on the right side of the screen. Figure B.11 shows CardioAnalyzer's GUI during myocardial perfusion assessment. Detail (1) concerns choosing the cardiac phase to use as a reference for defining myocardial segments. Only phases which have already been segmented appear in this list. Detail (2) refers to the list of cardiac phases to use during myocardial perfusion assessment. The selected phases will be pre-loaded in order to allow showing the animated sequence. The controls to enable/disable thick slices and to choose the method used (average, median, minimum and maximum intensity projection) and slice thickness are located in (3). In (4) the user can navigate along the different cardiac phases and enable the animated sequence with a customizable interval between phases. Finally, (5) is the interactive polar map which provides feedback of the current assessment state and allows its modification. This works as an alternative to the assessment performed by clicking on the lesions.

B.4 Conclusions

This chapter presented a description of CardioAnalyzer, a software application developed to support the work carried out concerning LV segmentation and analysis. It allowed easy integration of the different methods and tools developed and supported the different formal and informal evaluation tasks, in particular those described in chapters 8 and 10.

CardioAnalyzer has also provided a good framework to test and present the radiographers and clinicians with different features in order to assess what might be interesting for their work.

Considering the current features provided by CardioAnalyzer a few improvements can be performed. For example, concerning functional analysis, no data is stored after functional analysis



1. Cardiac phase to use as reference to define myocardial segments
2. Cardiac phases to use for analysis
3. Thick-slice definitions
4. Image navigation and animation
5. Interactive polar map

Figure B.11 – Graphical user interface showing myocardial perfusion assessment.

is performed. This means that the computation must be repeated each time the data is required for analysis. Having the data stored on disk would allow providing the user with, at least, a summary of the main outcomes when the patient is selected.

Also concerning functional analysis, when using the analysis board, the image view planes are not visible and when they are visible, only one of the parameters is shown. It might be interesting to allow users to add windows to the analysis board showing the desired view planes. For instance, they could have a short-axis view of the LV, side-by-side with the windows depicting the polar maps for the different parameters.

Bibliography

[1] O. Wink, H. Hecht, and D. Ruijters, "Coronary computed tomographic angiography in the cardiac catheterization laboratory: Current applications and future developments," *Cardiology Clinics*, vol. 27, no. 3, pp. 513–529, 2009.

[2] Y.-W. Wu, E. Tadamura, M. Yamamuro, S. Kanao, S. Okayama, N. Ozasa, M. Toma, T. Kimura, M. Komeda, and K. Togashi, "Estimation of global and regional cardiac function using 64-slice computed tomography: A comparison study with echocardiography, gated-spect and cardiovascular magnetic resonance," *International Journal of Cardiology*, vol. 128, no. 1, pp. 69–76, 2008.

[3] A. Mahnken, P. Bruners, S. Stanzel, R. Koos, G. Mühlenbruch, R. Günther, and P. Reinartz, "Functional imaging in the assessment of myocardial infarction: MR imaging vs. MDCT vs. SPECT," *European Journal of Radiology*, vol. 71, no. 3, pp. 480–485, 2009.

[4] N. Toussaint, T. Mansi, H. Delingette, N. Ayache, and M. Sermesant, "An integrated platform for dynamic cardiac simulation and image processing: Application to personalised tetralogy of fallot simulation," in *Proc. Eurographics Workshop on Vis. Comp. for Biomed.*, 2008.

[5] S. D. Olabarriaga and A. W. M. Smeulders, "Interaction in the segmentation of medical images: A survey," *Medical Image Analysis*, vol. 5, no. 2, pp. 127–142, 2001.

[6] M. D. Cerqueira, N. J. Weissmn, V. Dilsizian, A. K. Jacobs, S. Kaul, W. K. Laskey, D. J. Pennel, J. A. Rumberger, T. Ryan, and M. S. Verani, "Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart: A statement for healthcare professional from the cardiac imaging comitee of the council on clinical cardiology of the american heart association," *Circulation*, vol. 105, pp. 539–542, 2002.

[7] M. Budoff and J. Shinbane, *Cardiac CT Imaging - Diagnosis of Cardiovascular Disease*. Springer, 2006.

[8] K. Juergens and R. Fischbach, "Left ventricular function studied with MDCT," *European Radiology*, vol. 16, pp. 342–357, 2006.

[9] B. Desjardins and E. A. Kazerooni, "ECG-gated cardiac CT," *American Journal of Roentgenology*, vol. 182, pp. 993–1010, 2004.

[10] S. A. Lipson, *MDCT and 3D Workstations: A Pratical How-To Guide and Teaching File*. Springer, 2006.

[11] X. Alomar-Serrallach, E. Castillo-Gallo, and G. Pons-Lladó, *Basics and Performance of Cardiac Computed Tomography*. Springer, 2006.

[12] N. Hirai, J. Horiguchi, C. Fujioka, M. Kiguchi, H. Yamamoto, N. Matsuura, T. Kitagawa, H. Teragawa, N. Kohno, and K. Ito, "Prospective versus retrospective ECG-gated 64-detector coronary CT angiography: Assessment of image quality, stenosis, and radiation dose," *Radiology*, vol. 248, no. 2, pp. 424–430, 2008.

[13] W. P. Shuman, K. R. Branch, J. M. May, L. M. Mitsumori, D. W. Lockhart, T. J. Dubinsky, B. H. Warren, and J. H. Caldwell, "Prospective versus retrospective ECG gating for 64-detector CT of the coronary arteries: Comparison of image quality and patient radiation dose," *Radiology*, vol. 248, no. 2, pp. 431–437, 2008.

[14] O. Klass, M. Jeltsch, S. Feuerlein, H. Brunner, H.-D. Nagel, M. Walker, H.-J. Brambs, and M. Hoffmann, "Prospectively gated axial CT coronary angiography: preliminary experiences with a novel low-dose technique," *European Radiology*, vol. 19, no. 4, pp. 829–836, 2009.

[15] T. Flohr and S. Schaller, *Multidetector-row CT: Technical Principles*. Springer, 2004.

[16] H. K. Pannu, T. Flohr, F. M. Corl, and E. K. Fishman, "Current concepts in multi-detector row CT evaluation of the coronary arteries – principles, techniques, and anatomy," *Radiographics*, vol. 23, pp. S111–S125, 2003.

[17] T. Flohr, S. Schaller, K. Stierstorfer, H. Bruder, B. M. Ohnesorge, and U. J. Schoepf, "Multi-detector row CT systems and image-reconstruction techniques," *Radiology*, vol. 235, pp. 756–773, 2005.

[18] L. P. Lawler, H. K. Pannu, and E. K. Fishman, "MDCT evaluation of the coronary arteries, 2004: How we do it – data acquisition, postprocessing, display and interpretation," *American Journal of Roentgenology*, vol. 184, pp. 1402–1412, 2005.

[19] C. White and D. Kuo, "Chest pain in the emergency department: Role of multidetector CT," *Radiology*, vol. 245, no. 3, pp. 672–681, 2007.

[20] M. Kaus, J. Berg, J. Weese, W. Niessen, and V. Pekar, "Automated segmentation of the left ventricle in cardiac MRI," *Medical Image Analysis*, vol. 8, no. 3, pp. 245–254, 2004.

[21] M. Lynch, O. Ghita, and P. F. Whelan, "Automatic segmentation of the left ventricle cavity and myocardium in MRI data," *Computers in Biology and Medicine*, vol. 36, no. 4, pp. 389–407, 2006.

[22] R. Berbari, I. Bloch, A. Redheuil, E. D. Angelini, E. Mousseaux, and F. Frouin, "Automated segmentation of the left ventricle including papillary muscles in cardiac magnetic resonance images," in *Proc. 4^{th} Int. Conf. on Functional Imaging and Modeling of the Heart, LNCS 4466* (S. Berlin, ed.), pp. 453–462, 2007.

[23] J. Mille, R. Bone, P. Makris, and H. Cardot, "Segmentation and tracking of the left ventricle in 3D MRI sequences using an active surface model," in *Proc. 20th IEEE Int. Symp. on Computer-Based Medical Systems (CBMS 2007)*, pp. 257–262.

[24] H.-Y. Lee, N. Codella, M. Cham, M. Prince, J. Weinsaft, and Y. Wang, "Left ventricle segmentation using graph searching on intensity and gradient and a priori knowledge (lvGIGA) for short-axis magnetic resonance imaging," *Journal of Magnetic Resonance Imaging*, vol. 28, no. 6, pp. 1393–1401, 2008.

[25] M.-P. Jolly, H. Xue, L. Grady, and J. Guehring, "Combining registration and minimum surfaces for the segmentation of the left ventricle in cardiac cine MR images," in *Proc. 12th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2009, LNCS 5762*, pp. 910–918, 2009.

[26] J. S. Suri, "Computer vision, pattern recognition and image processing in left ventricle segmentation: The last 50 years," *Pattern Analysis & Applications*, vol. 3, no. 3, pp. 209–242, 2000.

[27] A. F. Frangi, W. Niessen, and M. Viergever, "Three-dimensional modeling for functional analysis of cardiac images: A review," *IEEE Transactions on Medical Imaging*, vol. 20, no. 1, pp. 2–25, 2001.

[28] W. Higgins, N. Chung, and E. Ritman, "Extraction of left-ventricular chamber from 3-D CT images of the heart," *IEEE Transactions on Medical Imaging*, vol. 4, no. 4, pp. 384–395, 1990.

[29] Z. Ma, J. M. R. S. Tavares, R. N. Jorge, and T. Mascarenhas, "A review of algorithms for medical image segmentation and their applications to the female pelvic cavity," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 13, no. 2, pp. 235–246, 2010.

[30] M.-P. Jolly, "Automatic segmentation of the left ventricle in cardiac MR and CT images," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 151–163, 2006.

[31] R. Redner and H. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, 1984.

[32] G. Muhlenbruch, M. Das, C. Hohl, J. Wildberger, D. Rinck, T. Flohr, R. Koos, C. Knackstedt, R. Gunther, and A. Mahnken, "Global left ventricular function in cardiac ct. evaluatioin of an automated 3D region-growing segmentation algorithm," *European Radiology*, vol. 16, pp. 1117–1123, 2006.

[33] J. Fleureau, M. Garreau, D. Boulmier, and A. Hernandez, "3D multi-object segmentation of cardiac MSCT imaging by using a multi-agent approach," in *Proc. 29th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society (EMBS 2007)*, pp. 6003–6006, 2007.

[34] J. Fleureau, M. Garreau, D. Boulmier, C. Leclercq, and A. Hernandez, "Segmentation 3D multi-objects d'images scanner cardiaques: Une approche multi-agents," *IRBM*, vol. 30, no. 3, pp. 104–113, 2009.

[35] J. Fleureau, M. Garreau, A. Simon, R. Hachemani, and D. Boulmier, "Assesment of global cardiac function in MSCT imaging using fuzzy connectedness segmentation," in *Proc. Computers in Cardiology 2008*, pp. 725–728, 2008.

[36] T. Okuyama, S. Ehara, N. Shirai, K. Sugioka, K. Ogawa, H. Oe, H. Kitamura, T. Itoh, K. Otani, T. Matsuoka, Y. Inoue, M. Ueda, T. Hozumi, and M. Yoshiyama, "Usefulness of three-dimensional automated quantification of left ventricular mass, volume, and function by 64-slice computed tomography," *Journal of Cardiology*, vol. 52, no. 3, pp. 76–284, 2008.

[37] D. Metaxas, T. Chen, X. Huang, and L. Axel, "Cardiac segmentation from MRI-tagged and CT images," in *Proc. 8th WSEAS International Conference on Computers, special session on Imaging and Image Processing of Dynamic Processes in Biology and Medicine*, 2004.

[38] X. Huang and D. Metaxas, "Metamorphs: Deformable shape and appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1444–1459, 2008.

[39] H. C. van Assen, M. G. Danilouchkine, M. Dirksen, J. H. Reiber, and B. P. Lelieveldt, "A 3-D active shape model driven by fuzzy inference: Application to cardiac CT and MR," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 5, pp. 595–605, 2008.

[40] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models – their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[41] O. Ecabert, J. Peters, C. Lorenz, J. von Berg, M. Vembar, K. Subramanyan, G. Lavi, and J. Weese, "Towards automatic full heart segmentation in computed-tomography images," in *Proc. Computers in Cardiology 2005*, pp. 223–226, 2005.

[42] O. Ecabert, J. Peters, H. Schramm, C. Lorenz, J. von Berg, M. Walker, M. Vembar, M. Olszewski, K. Subramanyan, G. Lavi, and J. Weese, "Automatic model-based segmentation of heart in CT images," *IEEE Transactions on Medical Imaging*, vol. 27, no. 9, pp. 1189–1201, 2008.

[43] D. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[44] Y. Zheng, B. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu, "Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features," *IEEE Transactions on Medical Imaging, Special Issue on Functional Imaging of the Heart*, vol. 27, no. 11, pp. 1668–1681, 2008.

[45] Y. Zheng, B. Georgescu, F. Vega-Higuera, and D. Comaniciu, "Left ventricle endocardium segmentation for cardiac CT volumes using an optimal smooth surface," in *Proc. SPIE Medical Imaging 2009: Image Processing*, vol. 7259, pp. 72593V–72593V–11, 2009.

[46] T. Schlosser, K. Pagonidis, C. Herborn, P. Hunold, K.-U. Waltering, T. Lauenstein, and J. Barkhausen, "Assessment of left ventricular parameters using 16-MDCT and new software for endocardial and epicardial border delineation," *American Journal of Roentgenology*, vol. 184, pp. 765–773, 2005.

[47] S. Raman, M. Shah, B. McCarthy, A. Garcia, and A. Ferketich, "Multi–detector row cardiac computed tomography accurately quantifies right and left ventricular size and function compared with cardiac magnetic resonance," *American Heart Journal*, vol. 151, no. 3, pp. 736–744, 2006.

[48] M. Dewey, M. Müller, F. Teige, and B. Hamm, "Evaluation of a semiautomatic software tool for left ventricular function analysis with 16-slice computed tomography," *European Radiology*, vol. 16, no. 1, pp. 25–31, 2006.

[49] N. Chaosuwannakit, P. Rerkpattanapipat, S. Wangsuphachart, and S. Srimahachota, "Reliability of the evaluation for left ventricular ejection fraction by ECG-gated multi-detector CT (MDCT): comparison with biplane cine left ventriculography," *Journal of the Medical Association of Thailand*, vol. 90, no. 3, pp. 532–538, 2007.

[50] J. Butler, M. Shapiro, D. Jassal, t. Neilan, J. Nichols, M. Ferencik, T. Brady, U. Hoffman, and R. Cury, "Comparison of multidetector computed tomography and two-dimensional transthoracic echocardiography for left ventricular assessment in patients with heart failure," *The American Journal of Cardiology*, vol. 99, no. 2, pp. 247–249, 2007.

[51] R. Fischbach, K. Juergens, M. Ozgun, D. Maintz, M. Grude, H. Seifarth, W. Heindel, and T. Wichter, "Assessment of regional left ventricular function with multidetector-row computed tomography versus magnetic resonance imaging," *European Radiology*, vol. 17, no. 4, pp. 1009–1017, 2007.

[52] M. Schipper, E. A. Kazerooni, P. Agarwal, S. Patel, J. Corbett, J. Jung, and K. Barber, "Left ventricular functional analysis with 16- and 64-row multidetector computed tomography: Comparison with gated single-photon emission computed tomography," *Journal of Computer Assisted Tomography*, vol. 33, no. 1, pp. 8–14, 2009.

[53] A. Sarwar, M. Shapiro, K. Nasir, K. Nieman, C. Nomura, T. Brady, and R. Cury, "Evaluating global and regional left ventricular function in patients with reperfused acute myocardial infarction by 64-slice multidetector CT: A comparison to magnetic resonance imaging," *Journal of Cardiovascular Computed Tomography*, vol. 3, no. 3, pp. 170–177, 2009.

[54] T. Okwuosa, C. Hampole, J. Ali, and K. Williams, "Left ventricular mass from gated SPECT myocardial perfusion imaging: Comparison with cardiac computed tomography," *Journal of Nuclear Cardiology*, vol. 16, no. 5, pp. 775–783, 2009.

[55] R. Cury, K. Nieman, M. Shapiro, K. Nasir, R. C. Cury, and T. Brady, "Comprehensive cardiac ct study: Evaluation of coronary arteries, left ventricular function, and myocardial perfusion – is it possible?," *Journal of Nuclear Cardiology*, vol. 14, no. 2, pp. 229–243, 2007.

[56] M. Aho, M. Gebregziabher, U. J. Schoepf, P. Suranyi, H. Lee, D. Gregg, P. Costello, and L. Zwerner, "Impact of right ventricular contrast attenuation on the accuracy of right ven-

tricular function analysis at cardiac multi-detector-row CT," *European Journal of Radiology*, vol. 73, no. 3, pp. 560–565, 2010.

[57] M. Müller, F. Teige, D. Schnapauff, B. Hamm, and M. Dewey, "Evaluation of right ventricular function with multidetector computed tomograpy: Comparison with magnetic resonance imaging and analysis of inter- and intraobserver variability," *European Radiology*, vol. 19, no. 2, pp. 278–289, 2009.

[58] S. Abbara, B. Chow, A. Pena, R. Cury, U. Hoffman, K. Nieman, and T. Brady, "Assessment of left ventricular function with 16- and 64-slice multi-.detector computed tomography," *European Journal of Radiology*, vol. 67, no. 3, pp. 481–486, 2008.

[59] W. K. Pratt, *Digital Image Processing*. Wiley, 4th ed., 2007.

[60] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[61] I.-C. Tsai, Y.-L. Huang, and K.-H. Kuo, "Left ventricular myocardium segmentation on arterial phase of multi-detector computed tomography," *Computerized Medical Imaging and Graphics*, vol. 36, no. 1, pp. 25–37, 2012.

[62] T. H. Kim, J. Hur, S. Kim, H. Kim, B. W. Choi, K. O. Choe, Y. Yoon, and H. Kwon, "Two-phase reconstruction for the assessment of left ventricular volume and function using retrospective ECG-gated MDCT: Comparison with echocardiography," *American Journal of Roentgenology*, vol. 185, pp. 319–325, 2005.

[63] Y. Kang, K. Engelke, and W. Kalender, "Interactive 3D editing tools for image segmentation," *Medical Image Analysis*, vol. 8, no. 1, pp. 35–46, 2004.

[64] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Computer Graphics (SIGGRAPH '87)*, vol. 21, no. 4, pp. 163–169, 1987.

[65] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit An Object-Oriented Approach to 3D Graphics*. Kitware, Inc., 4th ed., 2004.

[66] K. Babalola, B. Patenaude, P. Aljabar, J. Schnabel, D. Kennedy, W. Crum, S. Smith, T. Cootes, M. Jenkinson, and D. Rueckert, "An evaluation of four automatic methods of segmenting the subcortical structures in the brain," *NeuroImage*, vol. 47, pp. 1435–1447, 2009.

[67] R. Cárdenes, R. Luis-Garcia, and M. Bach-Cuadra, "A multidimensional segmentation evaluation for medical image data," *Computer Methods and Programs in Biomedicine*, vol. 96, no. 2, pp. 108–124, 2009.

[68] W. Yasnoff, J. Miu, and J. Bacus, "Error measures for scene segmentation," *Pattern Recognition*, vol. 9, pp. 217–231, 1977.

[69] K. Strasters and J. Gerbrands, "Three-dimensional segmentation using a split, merge and group approach," *Pattern Recognition Letters*, vol. 12, pp. 307–325, 1991.

[70] E. Pichon, A. Tannenbaum, and R. Kikinis, "A statistically based flow for image segmentation," *Medical Image Analysis*, vol. 8, pp. 267–274, 2004.

[71] A. Goumeidane, M. Khamadja, B. Belaroussi, H. Benoit-Cattin, and C. Odet, "New discrepancy measures for segmentation evaluation," in *Proc. IEEE Int. Conf. on Image Processing*, pp. 411–414, 2003.

[72] A. Popovic, M. de la Fuente, M. Engelhardt, and K. Radermacher, "Statistical validation metric for accuracy assessment in medical image segmentation," *International Journal of Computer Assisted Radiology and Surgery*, vol. 2, no. 3, pp. 169–181, 2007.

[73] T. Lasko, J. Bhagwat, K. Zou, and L. Ohno-Machado, "The use of receiver operating characteristic curves in biomedical informatics," *Journal of Biomedical Informatics*, vol. 38, pp. 404–415, 2005.

[74] G. Hripcsak and D. Heitjan, "Measuring agreement in medical informatics reliability studies," *Journal of Medical Informatics*, vol. 35, pp. 99–110, 2002.

[75] E. Harris, E. Donovan, J. Yarnold, C. Coles, and P. Evans, "Characterization of target volume changes during breast radiotherapy using implanted fiducial markers and portal imaging," *International journal of Radiation Oncology, Biology and Physics*, vol. 73, no. 3, pp. 958–966, 2009.

[76] H.-H. Chang, A. Zhuang, D. Valentino, and W.-C. Chu, "Performance measure characterization for evaluation neuroimage segmentation algorithms," *NeuroImage*, vol. 47, pp. 122–135, 2009.

[77] D. Hoaglin, F. Mosteller, and J. Tukey, *Understanding Robust and Exploratory Data Analysis*. John Wiley & Sons, 1983.

[78] D. G. Johnson, *Applied Multivariate Methods for Data Analysis*. Duxbury, 1998.

[79] J. Freund and G. Simon, *Statistics - a first course*. New Jersey: Prentice Hall, 6^{th} ed., 1995.

[80] S. Silva, B. Sousa Santos, J. Madeira, and A. Silva, "A 3D tool for leflt ventricle segmentation editing," in *Proc. Int. Conference on Image Analysis and Recognition (ICIAR 2010), LNCS 6112*, pp. 79–88, 2010.

[81] Y. Zhang, "A survey on evaluation methods for image segmentation," *Pattern Recognition*, vol. 29, no. 8, pp. 1335–1346, 1996.

[82] Y. Zhang, "A review of recent evaluation methods for image segmentation," in *Proc. Int. Symp. on Signal Processing and its Applicatins (ISSPA)*, pp. 148–151, 2001.

[83] H. Zhang, J. Fritts, and S. Goldman, "Image segmentation evaluation. a survey of unsupervised methods," *Computer Vision and Image Understanding*, vol. 110, pp. 260–280, 2008.

[84] I. Buvat, V. Charoy, F. Aubry, M. Pelegrini, G. Fakhri, C. Huguenin, H. Benali, A. Todd-Pokropek, and R. Di Paola, "The need to develop guidelines for evaluation of medical image processing procedures," in *Proc. SPIE Medical Imaging: Image Processing, vol. 3661*, pp. 1466–1477, 1999.

[85] J. Udupa, V. LeBlanc, Y. Zhuge, C. Imielinska, H. Schmidt, L. Currie, B. Hirsch, and J. Woodburn, "A framework for evaluating image segmentation algorithms," *Computerized Medical Imaging and Graphics*, vol. 39, no. 2, pp. 75–87, 2006.

[86] S. Vanbelle and A. Albert, "Agreement between an isolated rater and a group of raters," *Statistica Neerlandica*, vol. 63, no. 1, pp. 82–100, 2009.

[87] S. Bouix, M. Martin-Fernandez, L. Ungar, M. Nakamura, M.-S. Koo, R. McCarley, and M. Shenton, "On evaluating brain tissue classifiers without a ground truth," *NeuroImage*, vol. 36, pp. 1207–1224, 2007.

[88] S. Warfield, K. Zou, and W. Wells, "Simultaneous truth and performance level estimation (STAPLE): An algorithm for the validation of image segmentation," *IEEE Transactions on Medical Imaging*, vol. 23, pp. 903–921, 2004.

[89] M. Martin-Fernandez, B. Sylvain, L. Ungar, R. McCarley, and M. Shenton, "Two methods for validating brain tissue classifiers," in *Proc. 8th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI), LNCS 3749*, pp. 515–522, 2005.

[90] C. Hurkmans, M. Admiraal, M. van der Sangen, and I. Dijkmans, "Significance of breast boost volume changes during radiotherapy in relation to current clinical interobserver variations," *Radiotherapy and Oncology*, vol. 90, pp. 60–65, 2009.

[91] E. Kouwenhoven, M. Giezen, and H. Struikmans, "Measuring the similarity of target volume delineations independent of the number of observers," *Physics in Medicine and Biology*, vol. 54, pp. 2863–2873, 2009.

[92] E. Coche, M. Walker, F. Zech, and R. Crombrugghe, "Quantitative right and left ventricular functional analysis during gated whole-chest MDCT: A feasibility study comparing automatic segmentation to semi-manual contouring," *European Journal of Radiology*, vol. 74, no. 3, pp. 138–143, 2010.

[93] R. Steenbakkers, J. Duppen, I. Fitton, K. Deurloo, L. Zijp, E. Comans, A. Uitterhoeve, P. Rodrigus, G. Kramer, J. Bussink, K. Jaeger, J. Belderbos, P. Nowak, M. Herk, and C. Rasch, "Reduction of observer variation using matched CT-PET for lung cancer delineation: A three-dimensional analysis," *International Journal of Radiation Oncology, Biology and Physics*, vol. 64, no. 2, pp. 435–448, 2006.

[94] R. Steenbakkers, J. duppen, I. Fitton, K. deurloo, L. Zijp, A. Uitterhoeve, P. Rodrigus, G. Kramer, J. Bussink, K. Jaeger, J. Belderbos, A. Hart, P. Nowak, M. Herk, and C. Rasch, "Observer variation in target volume delineation of lung cancer related to radiation oncologist-computer interaction: A "big brother" evaluation," *Radiotherapy and Oncology*, vol. 77, pp. 182–190, 2005.

[95] S. Silva, J. Madeira, B. Sousa Santos, and C. Ferreira, "Inter-observer variability assessment of a left ventricle segmentation tool applied to 4D MDCT images of the heart," in *Proc. 33rd Int. Conf. of the IEEE Engineering in Medicine and Biology Society – EMBC 2011*, pp. 3411–3414, 2011.

[96] G. Pons-Lladó and R. Leta-Petracca, *Atlas of Non-Invasive Coronary Angiography by Multidetector Computed Tomography*. Springer, 2006.

[97] R. Ionasec, I. Voigt, B. Georgescu, Y. Wang, H. Houle, F. Vega-Higuera, N. Navab, and D. Comaniciu, "Patient-specific modeling and quantification of the aortic and mitral valves from 4-D cardiac CT and TEE," *IEEE Transactions on Medical Imaging*, vol. 29, no. 9, pp. 1636–1651, 2010.

[98] D. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler, "Challenges in visual data analysis," in *Proc. Information Visualization (IV06)*, pp. 9–16, 2006.

[99] D. Keim, "Information visualization and visual data mining," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 100–107, 2002.

[100] A. Palazzuoli, F. Cademartiri, M. Geleijnse, B. Meijboom, F. Pugliese, O. Soliman, A. Calabró, R. Nuti, and P. Feyter, "Left ventricular remodelling and systolic function measurement with 64 multi-slice computed tomography versus second harmonic echocardiography in patients with coronary artery disease: A double blind study," *European Journal of Radiology*, vol. 73, no. 1, pp. 82–88, 2010.

[101] S. Wesarg, "AHA conform analysis of myocardial function using and extending the toolkits ITK and VTK," in *Proc. Computer Assisted Radiology and Surgery (CARS 2005), International Congress Series*, vol. 1281, pp. 44–49, 2005.

[102] S. Kermani, M. Moradi, H. Abrishami-Moghaddam, A. Saneei, M. Marashi, and D. Shahbazi-Gahrouei, "Quantitative analysis of left ventricular performance from sequences of cardiac magnetic resonance imaging using active mesh model," *Computerized Medical Imaging and Graphics*, vol. 33, no. 3, pp. 222–234, 2009.

[103] S. Wesarg, "Monitoring treatment outcome: A visualization prototype for left ventricular transformation," in *Proc. FIMH 2011, LNCS vol. 6666*, pp. 121–128, 2011.

[104] M. Breeuer, "Comprehensive visualization of first-pass myocardial perfusion: The uptake movie and the perfusogram," in *Proc. Int. Soc. for Magnetic Resonance in Medicine*, 2002.

[105] T. Munzner, *Information Visualization: Human-Centered Issues and Perspectives, LNCS 4950*, ch. Process and pitfalls in writing infovis research papers, pp. 134–153. Springer, 2008.

[106] M. Keller, T. Grimm, *Knowledge and Information Visualization, LNCS 3426*, ch. The Impact of Dimensionality and Color Coding of Information Visualizations on Knowledge Acquisition, pp. 167–182. Springer-Verlag, 2005.

[107] M. Borkin, K. Gajos, A. Peters, D. Mitsouras, S. Melchionna, F. Rybicki, C. Feldman, and H. Pfister, "Evaluation of artery visualizations for heart disease diagnosis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, Dec. 2011.

[108] M. Plumlee and C. Ware, "Zooming versus multiple window interfaces: Cognitive costs of visual comparisons," *ACM Transactions on Computer-Human Interaction*, vol. 13, no. 2, pp. 1–31, 2006.

[109] S. Silva, B. Sousa Santos, and J. Madeira, "Using color in visualization: A survey," *Computers & Graphics*, vol. 33, no. 2, pp. 320–333, 2011.

[110] T. Kristensen, K. Kofoed, D. Moller, M. Ersboll, T. Kühl, P. Recke, L. Kober, M. Nielsen, and H. Kelbaek, "Quantitative assessment of left ventricular systolic wall thickening using multidetector computed tomography," *European Journal of Radiology*, vol. 72, no. 1, pp. 92–97, 2009.

[111] M. Prasad, A. Ramesh, P. Kavanagh, B. Tamarappoo, R. Nakazato, J. Gerlach, V. Cheng, L. tomson, D. Berman, G. Germano, and P. Slomka, "Quantification of 3D regional myocardial wall thickening from gated magnetic resonance images," *Journal of Magnetic Resonance Imaging*, vol. 31, pp. 317–327, 2010.

[112] R. Geest, A. Roos, E. van der Wall, and J. Reiber, "Quantitative analysis of cardiovascular MR images," *International Journal of Cardiac Imaging*, vol. 13, no. 3, pp. 247–258, 1997.

[113] M. Sénési, C. Defrance, E. Bollache, L. Perdrix, E. Mousseaux, and N. Kachenoura, "3D evaluation of myocardial systolic wall stress from cardiac magnetic resonance cine data," in *Proc. Computing in Cardiology 2010*, pp. 813–816, 2010.

[114] L. Zhong, Y. Su, S.-Y. Yeo, R.-S. Tan, N. Ghista, and G. Kassab, "Left ventricular regional wall curvedness and wall stress in patients with ischemic dilated cardiomyopathy," *Am J Physiol Heart Circ Physiol*, vol. 296, pp. H573–H584, 2009.

[115] S. Silva, N. Bettencourt, D. Leite, J. Rocha, M. Carvalho, J. Madeira, and B. Sousa Santos, "Myocardial perfusion analysis from adenosine-induced stress MDCT," in *Proc. Ib. Conf. on Pattern Recognition and Image Analysis (IbPRIA 2011), LNCS 6669*, pp. 717–725, 2011.

[116] R. Blankstein, L. Shturman, I. Rogers, J. Rocha-Filho, D. Okada, A. Sarwar, A. Soni, R. Loureiro, G. Feuchtner, H. Gewirtz, U. Hoffmann, W. Mamuya, T. Brady, and R. Cury, "Adenosine-induced stress myocadial perfusion imaging using dual-source cardiac computed tomography," *Journal of the American College of Cardiology*, vol. 54, no. 12, pp. 1072–1084, 2009.

[117] M. Salerno and G. Beller, "Noninvasive assessment of myocardial perfusion," *Circulation Cardiovascular Imaging*, vol. 2, pp. 412–424, 2009.

[118] B. Ali, E. Hsiao, and M. Carli, "Combined anatomic and perfusion imaging of the heart," *Current Cardiology Reports*, vol. 12, pp. 90–97, 2010.

[119] N. Al-Saadi and J. Bogaert, *Clinical Cardiac MRI*, ch. Myocardial Perfusion, pp. 143–172. Springer, 2005.

[120] D. Okada, B. Ghoshhajra, R. Blankstein, J. Rocha-Filho, L. Shturman, I. Rogers, H. Bezerra, A. Sarwar, H. Gewirtz, U. Hoffmann, W. Mamuya, T. Brady, and R. Cury, "Direct comparison of rest and adenosine stress myocardial perfusion CT with rest and stress SPECT," *Journal of Nuclear Cardiology*, vol. 17, no. 1, pp. 27–37, 2010.

[121] N. H. J. Pijls, P. van Schaardenburgh, G. Manoharan, E. Boersma, J.-W. Bech, M. vant Veer, F. Bär, J. Hoorntje, J. Koolen, W. Wijns, and B. de Bruyne, "Percutaneous coronary intervention of functionally nonsignificant stenosis: 5-year follow-up of the DEFER study," *Journal of the American College of Cardiology*, vol. 49, no. 21, pp. 2105–2111, 2007.

[122] J. Rocha-Filho, R. Blankstein, L. Shturman, H. Bezerra, D. Okada, I. Rogers, B. Ghoshhajra, U. Hoffmann, G. Feuchtner, W. Mamuya, T. Brady, and R. Cury, "Incremental value of adenosine induced stress myocardial perfusion imaging with dual-source CT at cardiac CT angiography," *Radiology*, vol. 254, pp. 410–419, 2010.

[123] B. Tamarappoo, D. dey, R. Nakazato, H. Shmilovich, T. Smith, V. Cheng, L. Thomson, S. Hayes, J. Friedman, G. Germano, P. Slomka, and D. Berman, "Comparison of the extent and severity of myocardial perfusion defects measured by CT coronary angiography and SPECT myocardial perfusion imaging," *Journal of the American College of Cardiology: Cardiovascular Imaging*, vol. 3, no. 10, pp. 1010–1019, 2010.

[124] N. Bettencourt, J. Rocha, N. Ferreira, G. Pires-Morais, M. Carvalho, D. Leite, B. Melica, L. Santos, A. Rodrigues, P. Braga, M. Teixeira, L. Simões, A. Leite-Moreira, S. Cardoso, E. Nagel, and V. Gama, "Incremental value of an integrated adenosine stress-rest MDCT perfusion protocol for detection of obstructive coronary artery disease," *Journal of Cardiovascular Computed Tomography*, vol. 5, pp. 392–405, 2011.

[125] G. Rodríguez-Granillo, M. Rosales, E. Degrossi, and A. Rodriguez, "Signal density of left ventricular myocardial segments and impact of beam hardening artifact: implications for myocardial perfusion assessment by multidetector CT coronary angiography," *The International Journal of Cardiovascular Imaging*, vol. 26, no. 3, pp. 345–354, 2010.

[126] N. Kachenoura, F. Veronesi, J. Lodato, C. Corsi, R. Mehta, B. Newby, R. Lang, and V. Mor-Avi, "Volumetric quantification of myocardial perfusion using analysis of multidetector computed tomography 3D datasets: Comparison with nuclear perfusion imaging," *European Radiology*, vol. 20, pp. 337–347, 2010.

[127] R. T. George, A. Arbab-Zadeh, J. Miller, K. Kitagawa, H.-J. Chang, D. Bluemke, L. Becker, O. Yousuf, J. Texter, A. Lardo, and J. Lima, "Adenosine stress 64– and 256–row detector computed tomography angiography and perfusion imaging: A pilot study evaluationg the transmural extent of perfusion abnormalities to predict atherosclerosis causing mocardial ischemia," *Circulation Cardiovascular Imaging*, vol. 2, pp. 174–182, 2009.

[128] C. Valdiviezo, M. Ambrose, V. Mehra, A. Lardo, J. Lima, and R. T. George, "Quantitative and qualitative analysis and interpretation of CT perfusion imaging," *Journal of Nuclear Cardiology*, vol. 17, no. 6, pp. 1091–1100, 2010.

[129] E. van der Wall, J. Schuijf, J. Bax, J. Jukema, and M. Schalij, "CT perfusion angiography; beware of artifacts!," *The International Journal of Cardiovascular Imaging*, vol. 26, no. 3, pp. 355–358, 2010.

[130] R. C. Cury, T. A. Magalhães, A. C. Borges, A. A. Shiozaki, P. A. Lemos, J. S. Júnior, J. C. Meneghetti, R. C. Cury, and C. E. Rochitte, "Dipyridamole stress and rest myocardial perfusion by 64-detector row computed tomography in patients with suspected coronary artery disease," *The American Journal of Cardiology*, vol. 106, no. 3, pp. 310–315, 2010.

[131] A. Nasis, S. Seneviratne, and T. DeFrance, "Advances in contrast-enhanced cardiovascular CT for the evaluation of myocardial perfusion," *Current Cardiovascular Imaging Reports*, vol. 3, pp. 372–381, 2010.

[132] A. So, J. Hsieh, J. Li, and T.-Y. Lee, "Beam hardening correction in ct myocardial perfusion measurement," *Physics in Medicine and Biology*, vol. 54, no. 10, p. 3031, 2009.

[133] K. Kitagawa, R. T. George, A. Arbab-Zadeh, J. Lima, and A. Lardo, "Characterization and correction of beam-hardening artifacts during dynamic volume CT assesment of myocardial perfusion," *Radiology*, vol. 256, no. 1, pp. 111–118, 2010.

[134] M. Termeer, J. Bescos, M. Breeuwer, A. Vilanova, F. Gerritsen, E. Groller, and E. Nagel, "Patient-specific coronary artery supply territory AHA diagrams," *Journal of Cardiovascular Magnetic Resonance*, vol. 11, no. Suppl 1, p. P103, 2009.

[135] C. McDonald, G. Schadow, M. Barnes, P. Dexter, M. Overhage, B. Mamlin, and J. McCoy, "Open-source software in informatics – why, how and what?," *International Journal of Medical Informatics*, vol. 69, pp. 165–174, 2003.

[136] P. Nagy, "Open source in imaging informatics," *Journal of Digital Imaging*, vol. 20, pp. 1–10, 2007.

[137] J. J. Caban, A. Joshi, and P. Nagy, "Rapid development of medical imaging tools with open-source libraries," *Journal of Digital Imaging*, vol. 20, no. 1, pp. 83–93, 2007.

[138] T. S. Yoo and M. J. Ackerman, "Open source software for medical image processing and visualization," *Communications of the ACM on Medical Image Modeling*, vol. 48, no. 2, pp. 55–59, 2005.

[139] I. Bitter, R. Uitert, I. Wolf, L. Ibáñez, and J.-M. Kuhnigk, "Comparison of four freely available frameworks for image processing and visualization that use itk," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 483–493, 2007.

[140] W. Liao, T. Deserno, and Spitzer, "Evaluation of free non-diagnostic DICOM software tools," in *Proc. SPIE Vol. 6919, Medical Imaging 2008: PACS and Imaging Informatics*, p. 691903.

[141] ITK, "Insight segmentation and registration toolkit," *http://www.itk.org/*, (online Mar 2009).

[142] L. Ibáñez, W. Schroeder, L. Ng, and J. Cates, *The ITK Software Guide*. Kitware, Inc., 2nd ed., 2005.

[143] T. S. Yoo and D. N. Metaxas, "Special issue: ITK - open science - combining open data and open source software: Medical image analysis with the insight toolkit," *Medical Image Analysis*, vol. 9, no. 6, 2005.

[144] D. Santos, E. Costa, and M. Gutierrez, "ITKFlowRun: A visual programming tool for ITK image filters," in *Proc. Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI), Technical Poster*.

[145] H. Le, R. Li, and S. Ourselin, "Towards a visual programming environment based on ITK for medical image analysis," in *Proc. Digital Imaging computing: Techniques and Applications (DICTA 2005)*, pp. 558–565.

[146] I. Wolf, M. Vetter, I. Wegner, and M. Nolden, "The medical imaging interaction toolkit (MITK) – a toolkit facilitating the creation of interactive software by extending VTK and ITK," in *Proc. of SPIE Vol. 5367, Medical Imaging 2004: Visualization, Image Guided Procedures and Display*, pp. 16–27.

[147] I. Wolf, M. Vetter, I. Wegner, T. Böttger, M. Nolden, M. Schöbinger, M. Hastenteufel, and T. Kunert, "The medical imaging interaction toolkit," *Medical Image Analysis*, vol. 9, no. 6, pp. 594–604, 2005.

[148] J. Tian, J. Xue, Y. Dai, J. Chen, and J. Zheng, "A novel software platform for medical image processing and analysing," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 6, pp. 800 – 812, 2008.

[149] A. Enquobahrie, P. Cheng, K. Gary, L. Ibáñez, D. Gobbi, F. Lindseth, Z. Yaniv, S. Aylward, J. Jomier, and K. Cleary, "The image-guided surgery toolkit IGSTK: An open source C++ software toolkit," *Journal of Digital Imaging*, vol. 20, pp. 21–33, 2007.

[150] K. Gary, L. Ibáñez, S. Aylward, D. Gobbi, M. Blake, and K. Cleary, "IGSTK: An open source software toolkit for image guided surgery," *IEEE Computer*, vol. 39, no. 4, pp. 46–53, 2006.

[151] T. Rudolph, M. Puls, C. Anderegg, L. Ebert, M. Broehan, A. Rudin, and J. Kowal, "MARVIN: A medical research application framework based on open source software," *Computer Methods and Programs in Biomedicine*, vol. 91, pp. 165–174, 2008.

[152] D. Gering, A. Nabavi, R. Kikinis, N. Hata, L. O'Donnell, W. Eric, L. Grimson, F. Jolesz, P. Black, and W. Wells III, "An integrated visualization system for surgical planing and guidance using image fusion and an open MR," *Journal of Magnetic Resonance Imaging*, vol. 13, no. 6, pp. 967–975, 2001.

[153] S. Pieper, B. Lorensen, W. Schroeder, and R. Kikinis, "The NA-MIC Kit: ITK, VTK, pipelines, grids and 3D Slicer as an open platform for the medical image computing community," in *Proc. IEEE Int. Symposium on Biomedical Imaging: Nano to Nano*, pp. 698–701, 2006.

[154] A. Loening and S. Gambhir, "AMIDE: A completely free system for medical imaging data analysis," *Journal of Nuclear Medicine*, vol. 42, no. 5, p. 192, 2001.

[155] A. Loening and S. Gambhir, "AMIDE: A free software tool for multimodality medical image analysis," *Molecular Imaging*, vol. 2, no. 3, pp. 131–137, 2003.

[156] X. Papademetris, M. Jackowski, N. Rajeevan, M. DiStaso, H. Okuda, R. Constable, and L. Staib, "BioImage Suite: An integrated medical image analysis suite: An update," *Insight Journal*, no. http://hdl.handle.net/1926/209, 2006.

[157] N. Toussaint, J.-C. Souplet, and P. Fillard, "MedINRIA: Medical image navigation and research tool by INRIA," in *Proc. of MICCAI'07 Workshop on Interaction in Medical Image Analysis and Visualization*, 2007.

[158] M. Koening, W. Spindler, and J. Rexilius, "Enbedding VTK and ITK into visual programming and rapid prototyping platform," in *Proc. of SPIE Vol. 6141, Medical Imaging 2006: Visualization, Image-Guided Procedures, and Display*, p. 6141O, 2006.

[159] C. Tietjen, K. Mühler, F. Ritter, O. Konrad, M. Hindennach, and B. Preim, "METK – the medical exploration toolkit," *Bildverarbeitung für die Medizin (BVM)*, pp. 407–411, 2008.

[160] D. Weinstein, S. Parker, J. Simpson, K. Zimmerman, and G. Jones, *Visualization in the SCIRun Problem-Solving Environment*, pp. 615–632. Elsevier, 2005.

[161] F. Prior, B. Erickson, and L. Tarbox, "Open source software projects of the caBIG in vivo imaging workspace software special interest group," *Journal of Digital Imaging*, vol. 20, pp. 94–100, 2007.

[162] C. Botha and F. Post, "Hybrid scheduling in the DeVIDE dataflow visualization environment," in *Proc. of Simulation and Visualization*, pp. 309–322, SCS Publishing House Erlangen.

[163] G. Lodygensky, M. Seghier, S. Warfield, T. C., S. Sizonenko, F. Lazeyras, and P. HÃ°ppi, "Intrauterine growth restriction affects the preterm infant's hippocampus," *Pediatric Research*, vol. 63, no. 4, pp. 438–443, 2008.

[164] N. Nakajima, J. Wada, T. Miki, J. Haraoka, and N. Hata, "Surface rendering-based virtual intraventricular endoscopy: Retrospective feasibility study and comparison to volume rendering-based approach," *NeuroImage*, vol. 37, pp. S89–S99, 2007.

[165] A. Raappana, J. Koivukangas, and T. Pirilä, "3D modeling-based surgical planning in transsphenoidal pituitary surgery – perliminary results," *Acta Oto-Laryngologica*, vol. 128, no. 9, pp. 1011–1018, 2008.

[166] V. Israel-Jost, P. Chouquet, S. Salmon, C. Blondet, E. Sonnendrücker, and A. Constantinesco, "Pinhole SPECT imaging: Compact projection/backprojection operator for efficient algebraic reconstruction," *IEEE Transactions on Medical Imaging*, vol. 25, no. 2, pp. 158–167, 2006.

[167] A. Naum, H. Tuunanen, E. Engblom, V. Oikonen, H. Sipilä, P. Iozzo, P. Nuutila, and J. Knuuti, "Simultaneous evaluation of myocardial blood flow, cadiac function and lung water content using $[^{15}O]\, H_2O$ and positron emission tomography," *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 34, pp. 563–572, 2007.

[168] V. Yushmanov, A. Kharlamov, F. Boada, and S. Jones, "Monitoring of brain potassium with rubidium flame photometry and mri," *Magnetic Resonance in Medicine*, vol. 57, pp. 494–500, 2007.

[169] Y. Zhu, X. Papademetris, A. Sinusas, and J. Duncan, "Integrated segmentation and motion analysis of cardiac MR images using a subject-specific dynamical model," in *Proc. IEEE Computer Vision and Pattern Recognition Workshops (CVPRW08)*, pp. 1–8, 2008.

[170] I. Laufer, M. Negishi, N. Rajeevan, C. Lacadie, and R. Constable, "Sensory and cognitive mechanisms of change detection in the context of speech," *Brain Structure and Function*, vol. 212, no. 5, pp. 427–442, 2008.

[171] F. Wang, J. Kalmar, E. Edmiston, L. Chepenik, Z. Bhagwagar, L. Spencer, B. Pittman, M. Jackowski, X. Papademetris, R. Constable, and H. Blumber, "Abnormal corpus callosum integrity in bipolar disorder: A diffusion tensor imaging study," *Biological Psychiatry*, vol. 64, no. 8, pp. 730–733, 2008.

[172] P.-L. Bazin, J. Cuzzocreo, M. Yassa, W. Gandler, M. McAuliffe, S. Bassett, and D. Pham, "Volumetric neuroimage analysis extensions for the mipav software package," *Journal of Neuroscience Methods*, vol. 165, no. 1, pp. 111–121, 2007.

[173] M. Calabrese, M. Filippi, M. Rovaris, I. Mattisi, V. Bernardi, M. Atzori, A. Favaretto, L. Barachino, L. Rinaldi, C. Romualdi, P. Perini, and P. Gallo, "Morphology and evolution of cortical lesions in multiple sclerosis. a longitudinal MRI study," *NeuroImage*, vol. 42, no. 4, pp. 1324–1328, 2008.

[174] M. Faustini, R. Neptune, R. Crawford, and S. Stanhope, "Manufacture of passive dynamic ankle-foot orthoses using selective laser sintering," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 2, pp. 784–790, 2008.

[175] J. Choi, B. Jeong, M. Rohan, A. Polcari, and M. Teicher, "Preliminary evidence for white matter tract abnormalities in young adults exposed to parental verbal abuse," *Biological Psychiatry*, vol. 65, no. 3, pp. 227–234, 2009.

[176] M. Yassa and C. Stark, "A quantitative evaluation of cross-participant registration techniques for mri studies of the medial temporal lobe," *NeuroImage*, vol. 44, no. 2, pp. 319–327, 2009.

[177] J.-M. Peyrat, M. Sermesant, X. Pennec, H. Delingette, C. Xu, R. McVeigh, and N. Ayache, "A computational framework for the statistical analysis of cardiac diffusion tensors:

Application to a small database of canine hearts," *IEEE Transactions on Medical Imaging*, vol. 26, no. 11, pp. 1500–1514, 2007.

[178] D. P. Barboriak, A. Padua, G. York, and J. MacFall, "Creation of DICOM–aware applications using ImageJ," *Journal of Digital Imaging*, vol. 18, no. 2, pp. 91–99, 2005.

[179] W. Burger and M. Burge, *Digital Image Processing: An Algorithmic Approach Using Java*. Springer, 2007.

[180] S. Koompairojn, A. Petkova, K. Hua, and P. Metarugcheep, "Semi-automatic segmentation and volume determination of brain mass-like lesion," in *Proc. 21st IEEE International Symposium on Computer-Based Medical Systems*, pp. 35–40, 2008.

[181] J.-M. Kuhnigk, V. Dicken, S. Zidowitz, L. Bornemann, B. Kuennerlen, S. Krass, H.-O. Peitgen, H.-H. Jend, W. Rau, and T. Achenbach, "New tools for computer assistance in thoracic CT. part 1: Functional analysis of lungs, lung lobes, and bronchopulmonary segments," *RadioGraphics*, vol. 25, no. 2, pp. 525–536, 2005.

[182] B. Preim and S. Oeltze, *3D Visualization of Vasculature: An Overview*, pp. 39–59. Springer, 2007.

[183] S. Oeltze, H. Doleisch, H. Hauser, P. Muigg, and B. Preim, "Interactive visual analysis of perfusion data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1392–1399, 2007.

[184] N. Dedual, B. Johnson, and G. Chen, "Visualization of 4D computed tomography datasets," in *Proc. IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 120–123, 2006.

[185] K. Maheshwari, S. Olabarriaga, C. Botha, J. Snel, J. Alkemade, and A. Belloum, "Problem solving environment for medical image analysis," in *Proc. 20th International Symposium on Computer-Based Medical Systems (CBMS'07)*, pp. 165–170, 2007.

[186] G. Grevera, J. Udupa, D. Odhner, Y. Zhuge, A. Souza, T. Iwanaga, and S. Mishra, "Cavass: A computer-assisted visualization and analysis software system," *Journal of Digital Imaging*, vol. 20, no. 1, pp. 101–118, 2007.

[187] J. Grevera, G. Udupa, D. Odhner, Y. Zhuge, and A. Souza, "The architecture and performance of CAVASS," in *Proc. of SPIE Medical Imaging 2008: Visualization, Image Guided Procedures, and Modeling, Vol. 6918*, p. 691833, 2008.

[188] I. Sommerville, *Software Engineering*. Addison Wesley, 7^{th} ed., 2006.

[189] D. Mayhew, *The Usability Engineering Lifecycle – A Practitioners Handbook for User Interface Design*. Prentice Hall, 1999.

[190] A. Dix, J. Finlay, G. Abowd, and R. Beale, *Human-Computer Interaction*. Prentice Hall, 3^{rd} ed., 2004.

Index

adaptive detector array, 20
anular tomographic regions, 13
aorta, 11
artery, 11

beta-blockers, 25
bull's eye, 13

C-Factor, 93

Dice coefficient, 90

ECG, 12
ECG gating, 24
ECG triggering, 23
ejection fraction, 15
electrocardiogram, 12
end-diastole, 12
end-systole, 12
endocardium, 11
epicardium, 11
External Distortion Rate (EDR), 92

FOM (figure of merit) (Strasters), 91

Hausdorff distance, 91
helical acquisition, 19
horizontal long axis (plane), 13

Internal Distortion Rate (IDR), 92
ITK, 63, 176

Jaccard, 90
JCd (Cárdenes), 92

left ventricle, 11

MeVisLab, 63, 185
mitral valve, 11
myocardial segment, 13
myocardium, 11

neighbourhood iterator (voxel), 77

papillary muscles, 11
polar map, 13
prospective scanning, *see* ECG triggering
pulmonary arteries, 11

retrospective scanning, *see* ECG gating
right ventricle, 11
RR interval, 12

scanogram, *see* topogram
scout scan, *see* topogram
short axis (plane), 13
spiral acquisition, 19
stroke volume, 15

Tanimoto metric, 90
topogram, 27
tricuspid valve, 11

vein, 11
vertical long axis (plane), 13

Yasnoff discrepancy measure, 91

This edition is limited to fifteen copies. Nine copies *hors commerce*, with colour figures and binding by Tipografia Minerva, numbered from I to IX and signed by the author; and six copies with black and white figures, without numbering, for reviewing purposes.

Aveiro, February 1^{th}, 2012